الفصل الثاني
المكدس Stack

2.1 المقدمة:

هياكل البيانات هي تجميع لعناصر تنظم بخط مستقيم (مثل المصفوفات وربما القوائم الموصول) ..المكدس والطابور هي هياكل بيانات خطية حيث ان البيانات منظمة بشكل خطي, ويمتاز المكدس بتطبيقاتة المتعددة التي من شانها ان تسهل العمل على الحاسوب وتزيد من كفائتة.

2.2 **المكدس**: هو طريقة خاصة لخزن البيانات والتعامل معها, وهو نسخة محددة من المصفوفات, العناصر الجديدة ممكن ان تضاف الى المكدس او تحذف من المكدس من نهاية واحدة, فهو مستودع بيانات حيث ان اخر عنصر يضاف سيخرج اولا.
ويتبع المكدس هيكلية الداخل اولا خارج اخيرا ويمكن ان يمثل ماديا كمصفوفة ويدعى حينها هياكل البيانات المستقرة ***static data structure*** **وعندها سيرث جميع صفات المصفوفات,** او من الممكن ان يمثل كقائمة موصولة ويدعى هياكل البيانات الديناميكية ***dynamic data structure*** **وسيمتلك جميع خواص القوائم الموصولة وبغض النظر عن طريقة تمثيل المكدس فان عمليات الاضافة والحذف تحدث في اعلى المكدس فقط.**

شكل 2.1 : نماذج تصف المكدس

يستخدم عادة المكدس هياكل البيانات عند كتابة الشفرة. مبدأ المكدس في الحقيقة بسيط, والذي يجعلة حتى ابسط لكتابة شفرتة. افرض هذه الحالة, لديك خمسة كتب موضوعة على طاولة فوق بعضها البعض (شكل 2.1) فاذا رغبت ان تضيف كتاب سادس لها, فماذا تعمل بكل تاكيد ستضع الكتاب السادس فوق الكتب الخمس, الان ماذا لو اردت ان تسحب الكتاب الثالث من الكتب الست التي لديك, هنا عليك ان تزيل الكتاب الأعلى ثم الذي بعده وهكذا لحين ان يصبح الكتاب الثالث في **الأعلى** ليتسنى لك الحصول

عليه ومن ثم اعادة الكتب التي ازلتها الى مكانها وحسب ترتيبها اي ان تضعها كتاب بعد الاخر (فوق او **أعلى** الكتب الاخرى).

لاحظ هنا, اني اشرت الى كلمة الأعلى بحروف غامقة, نعم لان أعلى المكدس هو اكثر كلمة مهمة في التعامل مع المكدس. البيانات تخزن في المكدس بحيث ان اضافة البيانات لايسمح بها الا من الأعلى فقط. ان اضافة وحذف البيانات لايسمح به الا من أعلى المكدس فقط.

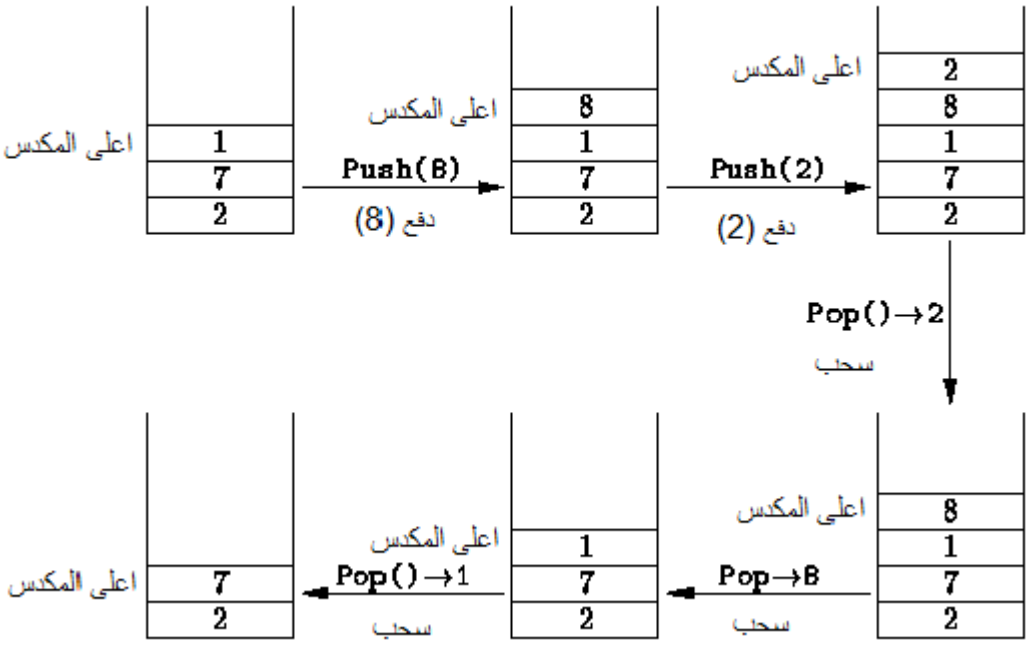
العمليات الأساسية التي تجري على المكدس هي:

الدفع او الاضافة Push: وهي عملية اضافة عنصر جديد الى أعلى المكدس.

السحب او الحذف Pop: وهي عملية سحب عنصر من أعلى المكدس.

هذه العمليات (الاضافة والحذف) موضحة بالشكل 2-2 حيث نلاحظ ان الاضافة تتم من اعلى المكدس وعنصر واحد في كل مرة, وكذلك فان الحذف يتم من اعلى المكدس.. لاحظ هنا في حالة الاضافة يجب ان تحدد القيمة المضافة, مع مراعاة ان جميع عناصر المكدس هي من ذات النوع, اما في حالة الحذف فاننا نحذف العنصر باعلى المكدس بغض النظر عن قيمتة ويمكن فقط تقليل قيمة مؤشر اعلى المكدس بمقدار واحد ليتم حذف لاعنصر دون الحاجة لمعرفة قيمتة.

كذلك لاحظ وجود مؤشر يشير الى موقع اعلى عنصر بالمكدس (يمثل عنوان العنصر الذي في اعلى المكدس بالذامرة المخصصة للمكدس).



شكل (2-2): يوضع عمليات الاضافة والحذف من المكدس

2.3 تطبيقات المكدس:

المكدس له تطبيقات كثيرة. ادناة بعض تطبيقات المكدس:

2.3.1 عند تنفيذ اي برنامج فان المعالج سيكون له الدور الاساس بعملية التنفيذ, لذلك اذا حدث وان تم استدعاء دالة معينة في البرنامج فان المعالج سيتوقف عن اكمال تنفيذ البرنامج حسب تسلسل

الايعازات او الاوامر في البرنامج ويتفرع لاكمال تنفيذ الدالة ثم العودة الى ذات الموقع الذي تفرع منه لاكمال التنفيذ حسب تسلسل الايعازات, في هذه الحالة فان المعالج يجب ان تكون له الامكانية للعودة الى البرنامج في الموقع الذي تفرع منه لتنفيذ الدالة التي تم استدعائها, لذلك فان العنوان الحالي للبرنامج (اي الايعاز الذي كان من المفروض ان ينفذ قبل ان يتفرع المعالج الى تنفيذ الدالة سيدفع الى المكدس ليتم حفظة, وعندما ينتهي المعالج من تنفيذ الدالة, فان العنوان الذي تم خزنة سيتم سحبة من المكدس ليوجهة المعالج اليه لاكمال التنفيذ.

2.3.2 والمكدس يدعم طريقة الاستدعاء الذاتي. المكدس يستخدم ايضا بواسطة المترجمات بعملية معالجة او تقييم التعابير وتوليد شفرة لغة الماكنة, كذلك يستخدم لخزن عناوين الاعادة بسلسلة من طرق الاستدعاءات خلال تنفيذ البرنامج.

2.3.3 يستخدم المكدس بشكل كبير في انظمة الحواسيب الحديثة, مثال الحاسبات الشخصية الحديثة تستخدم المكدس على مستوى معمارية الحاسوب, حيث يستخدم في التصميم الاساس لنظام التشغيل لاغراض تنفيذ المقاطعة واستدعاء دوال نظام التشغيل, ومن بين الاستخدامات الاخرى, فان المكدس يستخدم لتشغيل او تنفيذ ماكنة جافا الافتراضية Java virtual machine ولغة جافا نفسها لديها صنف يسمى المكدس, والذي من الممكن استخدامة من المبرمج. المكدس في كل مكان, انظمة الحاسوب التي تعتمد على المكدس هي واحدة من التي تخزن المعلومات بشكل مؤقت في المكدس بدلا من مسجلات وحدة المعالجة المركزية.

2.3.4 تطبيق اخر, نفرض انك تقوم بعملية جمع أعداد كبيرة جدا, على سبيل المثال افرض انك ترغب بجمع العددين

- 353,120,457,764,910,452,008,700
- 234,765,000,129,654,080,277

اولا لاحظ التالي انه من الصعب تمثيل اي من العددين كمتغير من نوع الاعداد الصحيحة (integer) لان هذا المتغير لايمكنة من استيعاب هذا العدد الكبير. هذه المشكلة بالامكان حلها وذلك بمعاملة العدد كسلسلة رمزية من الارقام, ويتم خزنهم بمكدسين, بعدها تجرى عملية الاضافة او الجمع بسحب ارقام من المكدسين.

2.3.5 ايجاد نتيجة تعبير حسابي: المكدس مفيد لايجاد نتيجة تعبير حسابي, افرض لديك التعبير التالي

5 * 3 +2 + 6 * 4

ممكن ايجاد نتيجة التعبير وذلك بايجاد, اولا نتيجة حاصل ضرب 5 في 3 , بعد ذلك تخزن النتيجة في متغير مثلا A, ثم قم باضافة الرقم 2 الى المتغير A واخزن النتيجة بالمتغير A. الان اجري عملية الضرب للرقمين 6 في 4 واخزن الناتج بالمتغير B. عند ذلك ستنتهي العملية باضافة A الى B واترك النتيجة النهائية بالمتغير A .

A = 15	(5 3 *)
A = 15 + 2	(15 2 +)
= 17	
B = 6 * 4	(6 4 *)
= 24	
A = 17 + 24	(17 24 +)
= 41	

بالامكان كتابة هذه العمليات المتسلسلة كمايلي:

5 3 * 2 + 6 4 * +

ويعرف هذا التدوين (تعبير العلامة اللاحقة postfix) ويتم تقييمها على النحو المبين أعلاه. سنوضح قريبا كيف يمكن إنشاء هذا التدوين باستخدام المكدس.

2.4 التعابير الجبرية Algebraic Expression

- التعبير الجبري: هو ترتيب رياضي يحتوي على معاملات وعوامل رياضية.
- المعاملات: هي كميات (وحدة بيانات) والتي تجرى عليها العمليات الرياضية.
  والمعاملات ممكن ان تكون متغيرات مثل x, y, z او ثوابت مثل 1- ,2, 34, 12 ..الخ
- العوامل الرياضية: هي رموز تدل على عمليات رياضية او منطقية بين المعاملات, مثال على بعض العوامل الشائعة =< ,^ ,/ ,* ,- ,+
  اذا مااخذنا هذين التعريفين للعوامل والمعاملات, فاننا يمكننا الان ان نكتب تعبير مثل

x + y * z

واقعا هناك ثلاث أنواع من التعابير الرياضية. وهي:

2.4.1 تعبير العلامة القياسية infix: وهذا هو النوع الذي اعتدنا على التعامل معه في الرياضيات منذ الايام الاولى للمدرسة, حيث يكون موضع العامل الرياضي (العلامة الرياضية) بين المعاملين الذين من المفروض ان تجرى عليهم العملية الرياضية وفقا لهذا العامل, مثل

A + B

2.4.2 تعبير العلامة اللاحقة Postfix: في هذا التعبير يكون العامل الرياضي تابع او لاحق الى المعاملين اي بعد المعاملين الذين من المفروض ان تجرى عليهم العملية الرياضية وفقا لهذا العامل, مثل

A B +

2.4.3 تعبير العلامة السابقة Prefix: هنا يكون مكان العامل الرياضي قبل المعاملين الذين من المفروض ان تجرى عليهم العملية الرياضية وفقا لهذا العامل, مثل

+ A B

لاحظ ان تعبير العلامة القياسي لايمكن ان تستخرج نتيجة (يقيم evaluate) باستخدام ترتيب العوامل الرياضية من اليسار الى اليمين داخل التعبير. على كل حال, فان العلامات الرياضية داخل تعبير العلامة اللاحقة دائما تكون في مكان التقييم الصحيح, عليه فان التقييم او الحساب لتعبير العلامة القياسي يعمل او ينجز بخطوتين .. الاولى هو تحويل تعبير العلامة القياسية الى مايكافئة من تعبير العلامة اللاحقة الخطوة الثانية تتضمن حساب تعبير العلامة اللاحقة. سوف نرى ذلك في هذا الفصل, كيف ان المكدس مفيد بالتعامل مع كلا الخطوتين, دعنا اولا نختبر العملية الاساسية للتحويل من تعبير العلامة القياسية الى تعبير العلامة اللاحقة .

2.5 تحويل تعبير العلامة القياسية الى تعبير العلامة اللاحقة

2.5.1 اسباب استخدام تعبير العلامة اللاحقة وتعبير العلامة السابقة

بالتاكيد فان القاريء ربما يتسائل لماذا نستخدم هذا الشكل الغريب من تعابير العلامة السابقة والعلامة اللاحقة.

عندما يكون لدينا تعبير علامة قياسية بسيط, فسيكون من المفاجأ ان تعرف ان تعبير العلامة القياسية ليس بسيط كما يبدو خصوصا عندما ترغب بايجاد نتيجتة. لغرض حساب قيمة تعبير العلامة القياسية فانك تحتاج الى الاعتماد على افضلية العوامل والصفات المشتركة معة, مثال:

التعبير 4*5+3 يحسب على انه 32 وهو يعني 4*(5+3), او يحسب 23 بمعنى (4*5)+3 .

ولحل هذه المشكلة فان الاسبقيات او الافضليات للعوامل تم تحديدها, حيث ان اسبقية العوامل تتحكم بترتيب حساب القيم للعمليات الحسابية المختلفة ضمن التعبير الحسابي الواحد. العامل ذو الاسبقية الاعلى يتم حسابة او تطبيقة قبل العامل ذو الاسبقية الاوطأ.

فتعبير العلامة القياسي من الصعب تحليلة وذلك:

- لانه يحتاج الى تحديد افضلية العوامل, المحددات, وكسر التعادل (اي عندما يكون عاملين بنفس الافضلية), وهذا يجعل تقييم الحاسوب للتعبير صعب اكثر من اللازم.
- بينما تعبير العلامة السابقة وتعبير العلامة اللاحقة لايعتمد على افضلية العوامل, المحددات, وكسر التعادل. لذا فان تقييم التعابير بهذه الصيغ يكون اسهل.

جدول 2.1 : يوضح اسبقية العوامل الرياضية والمنطقية في الحاسوب

العوامـــــــل	الافضلية	الملاحظـــــــــــــات
كل العوامل الاحادية, ^	**6**	**^ تعني الرفع الى اس معين, اما العوامل الاحادية فهي تعني العوامل الرياضي التي تسبق معامل واحد فقط مثل (-7).**
/ , * , %	**5**	**(%) تشير الى باقي القسمة**
+ , - الثنائية	**4**	**الثنائية تعني انها تستخدم معاملين واحد قبلها والاخر بعدها مثل (4+ 12).**
==, <, >, <=, >=, !=	**3**	**تمثل العلامات المنطقية فقط**
Not	**2**	
&&, ||	**1**	**علامتي and, or**
=	**0**	**المساواة التي تستخدم باسناد قيمة معينة الى متغير**

2.5.2 خوارزمية تحويل تعبير العلامة القياسي الى تعبير العلامة اللاحقة

1. يعامل تعبير العلامة القياسي على انه سلسلة من الرموز (معاملات وعوامل رياضية) ويتم فحصها او قراءة هذه الرموز واحد بعد الاخر من اليسار الى اليمين.
2. يتم استخدام المكدس كخزان وسطي يكون ابتداءا فارغ, وسلسلة للمخرجات تكون بداية خالية من اي رمز.

3. افحص العنصر اللاحق بالمدخلات (اي ضمن تعبير العلامة القياسية).
4. اذا كان هذا العنصر معامل (متغير او ثابت) يتم وضعة في سلسلة المخرجات مباشرة.
5. اذا كان قوس مفتوح يتم دفعة الى المكدس.
6. اذا كان عامل رياضي قم بمايلي:
   a) اذا كان المكدس فارغ ادفع العامل الى المكدس. اذهب الى الخطوة 7
   b) اذا كان ماموجود باعلى المكدس هو قوس مفتوح ادفع العامل الى المكدس. اذهب الى الخطوة 7
   c) اذا كان هذا العامل له اسبقية اعلى من العامل في اعلى المكدس ادفع هذا العامل الى المكدس. اذهب الى الخطوة 7
   d) اذا كان هذا العامل له اسبقية اوطأ من العامل الموجود في اعلى المكدس, في هذه الحالة اسحب العامل من اعلى المكدس واخرجة ليتم وضعة في سلسلة المخرجات, اذهب الى الخطوة a في بداية الفقرة 6
7. اذا كان قوس مغلق اسحب العوامل من المكدس ويتم وضعهم في سلسلة المخرجات واحد بعد الاخر, لحين تصل الى القوس المفتوح. عملية السحب ستهمل القوس المفتوح.
8. اذا كان هناك مزيد من المدخلات اذهب الى الخطوة 1
9. اذا لم يكن هناك مزيد من المدخلات, اسحب العوامل المتبقية في المكدس واضفهم الى سلسلة المخرجات

2.5.3 امثلة لتحويل تعبير العلامة القياسي الى تعبير العلامة اللاحقة (هنا باستخدام الاسبقيات وليس المكدس)

مثال 1: تعبير العلامة القياسية a + b * c
اسبقية عامل الضرب اكبر من اسبقية عامل الجمع
a + (b * c) // تحويل الضرب الى تعبير العلامة اللاحقة
a + (b c *) // تحويل الجمع الى تعبير العلامة اللاحقة
a (b c *) + // ازالة الاقواس

شكل تعبير العلامة اللاحقة a b c * +
لاحظ عدم الحاجة الى الاقواس في تعبير العلامة اللاحقة

مثال 2: تعبير العلامة القياسية (A + B) * C
لاحظ هنا ان اسبقية الاقواس اكبر من اسبقية العمليات الاخرى (اكبر من اسبقية عامل الضرب)
(A B +) * C // تحويل الجمع الى تعبير العلامة اللاحقة
(A B +) C * // تحويل الضرب الى تعبير العلامة اللاحقة
AB+C* // ازالة الاقواس

شكل تعبير العلامة اللاحقة A B + C *
لاحاجة للاقواس باي شكل من الاشكال

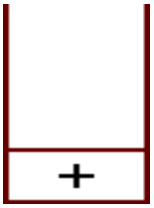
مثال 3: تعبير العلامة القياسية a + ((b * c) / d)
اسبقية علامة الضرب وعلامة القسمة متساويتان لذلك تكون الاسبقية للموجودة على اليسار, لكن في هذا المثال الاسبقية للاقواس
a + ((b c *) /d) // تحويل الضرب الى تعبير العلامة اللاحقة

a + (b c * d /) // تحويل القسمة الى تعبير العلامة اللاحقة
+)b c * d /(a // تحويل الجمع الى تعبير العلامة اللاحقة
abc*d/+ // ازالة الاقواس

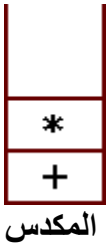
مثال 4: مثال توضيحي لكيفية تنفيذ خوارزمية تحويل تعبير العلامة القياسية بالستخدام المكدس:

تعبير العلامة القياسية a+b*c-d

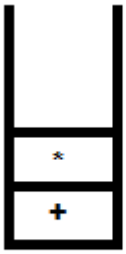
ابتداءا فان المكدس يكون فارغ وسلسلة التعبير اللاحق ليس فيها اي رموز. الان, الرمز الاول الذي يتم فحصه هو 'a' والذي ستتم اضافتة الى سلسلة تعبير العلامة اللاحقة. الرمز الثاني الذي تم فحصة هو علامة الجمع '+' وهذا بالطبع عامل رياضي وسيتم اضافتة الى المكدس.

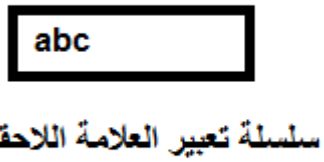


الرمز التالي الذي سيتم قرائتة هو الرمز 'b' والذي سيوضع في سلسلة تعبير العلامة اللاحقة. بعدها يكون الدور للرمز الذي سيتم قرائته هو علامة الضرب '*' والذي هو عامل رياضي. الان هذا يجب ان يودع في المكدس ويجب ان نلاحظ ان اعلى المكدس يحتوي العامل الرياضي '+' والذي هو ذو اسبقية اقل من عامل الضرب, لذلك فان عامل الضرب سيدفع الى اعلى المكدس.



الرمز اللاحق هو 'c' والذي سيوضع في تعبير العلامة اللاحقة.





الرمز الاخر الذي سيتم قرائته هو علامة الطرح '-' وهذا عامل رياضي المفروض ان يوضع في المكدس, لاحظ ان اعلى المكدس يحتوي علامة الضرب '*' والذي هو ذو اسبقية اعلى من عامل الطرح ونظرا لعدم امكانية ان يوضع عامل رياضي ذو اسبقية دنيا فوق عامل رياضي ذو اسبقية اعلى في المكدس لذا سيتم سحب العامل '*' من المكدس ويضاف الى سلسلة تعبير العلامة اللاحقة. حتى الان فان المكدس ليس فارغ. الان بعد سحب علامة الضرب فان علامة الجمع '+' هي في اعلى المكدس وهذا العامل مساوي بالاسبقية لعامل الطرح, ولذا فان عاملين متساويين بالاسبقية لايمكن ان يكونا احدهما فوق الاخر مباشرة بالمكدس لذا فان العامل '+' يسحب ايضا من المكدس ويوضع في سلسلة تعبير العلامة اللاحقة, عند ذاك يتم دفع العامل '-' الى المكدس.

المكدس

abc*+

سلسلة تعبير العلامة اللاحقة

الرمز الذي سيقرا الان هو الرمز 'd' والذي سيضاف الى تعبير العلامة اللاحقة, وبعد قراءة الرمز d نكون قد قرأنا جميع الرموز في تعبير العلامة القياسية وبهذا وصلنا الى حالة هي عدم وجود اي رمز اخر لم يتم قرائتة وعلية فان جميع العناصر الموجودة في المكدس سيتم سحبها واحد بعد الاخر, وتتم اضافتها الى سلسلة تعبير العلامة اللاحقة بالتعاقب. فيتم هنا سحب عامل الطرح ويضاف الى سلسلة تعبير العلامة اللاحقة. لذا بعد ان يتم قراءة جميع الرموز فان المكدس وسلسلة تعبير العلامة اللاحقة سيكونان كما في الشكل ادناة:

المكدس

abc*+d-

سلسلة تعبير العلامة اللاحقة

النتيجة النهائية ستكون:

- abc*+d- سلسلة تعبير العلامة اللاحقة

تمرين 1: حول تعبير العلامة القياسية في الامثلة التالية الى تعبير العلامة اللاحقة باستخدام الاسبقيات واستخدام المكدس. (*الاجوبة في نهاية الفصل*)

1. (a + b) * (c – d)
2. a – b / (c + d * e)
3. ((a + b) * c – (d – e))/(f + g)

مثال 5: مطلوب تحويل تعبير العلامة القياسي 2*3/(2-1)+5*3 الى تعبير العلامة اللاحقة باستخدام المكدس؟

2*3/(2-1)+5*3		
سلسلة المخرجات تعبير العلامة اللاحقة	**المكدس**	**عناصر تعبير العلامة القياسية**
2	فارغ	2
2	*	*
23	*	3
23*	/	/
23*	/(	(
23*2	/(	2
23*2	/(-	-
23*21	/(-	1
23*21-	/	)
23*21-/	+	+
23*21-/5	+	5
23*21-/5	+*	*
23*21-/53	+*	3
23*21-/53*+	فارغ	
23*21-/53*+	تعبير العلامة اللاحقة	

2.6 حساب (ايجاد نتيجة) تعبير العلامة اللاحقة

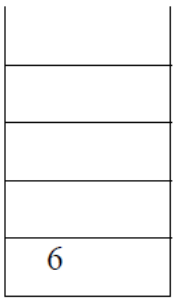
بالامكان حساب نتيجة تعبير العلامة اللاحقة باستخدام المكدس.

1. ابدأ بمكدس فارغ.
2. تعبير العلامة اللاحقة يتكون عادة من معاملات وعوامل رياضية, بالامكان اعتبارها كسلسلة من الرموز, يتم قرائتها او فحصها من اليسار الى اليمين تباعا (رمز كل مرة).
3. يتم قراءة رمز واحد من السلسلة في كل مرة.
4. اذا كان الرمز هو معامل يتم دفعة الى داخل المكدس.
5. اذا كان الرمز عامل رياضي, عندها يتم مايلي:

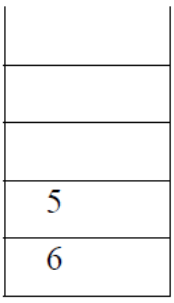
a) سحب اثنين من المعاملات من اعلى المكدس (اي المعامل الاعلى في المكدس والمعامل الذي يليه).
b) اجراء العملية الحسابية على المعاملين الذين تم سحبهما ووفقا للعامل الرياضي الذي تم قرائتة في الخطوه 5 .
c) يتم دفع النتيجة المستحصلة من الخطوة b الى داخل المكدس (توضع عادة في اعلى المكدس)

6. اذا لم تكن السلسلة فارغة الانتقال الى الخطوة 2 .

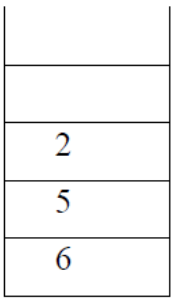
مثال 6: مثال توضيحي لايجاد نتيجة تعبير العلامة اللاحقة التالي: **6 5 2 3 + 8 * + 3 + ***



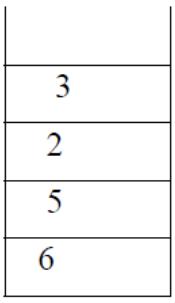
1



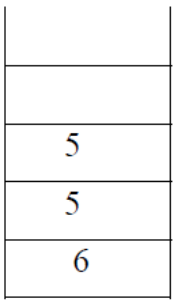
2



3

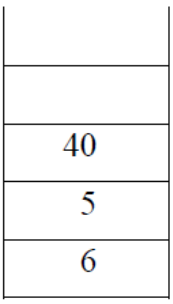


4

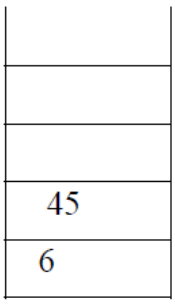


5

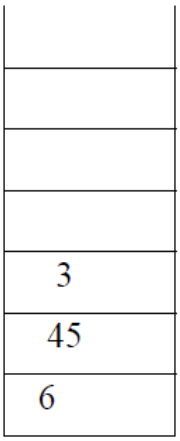
6



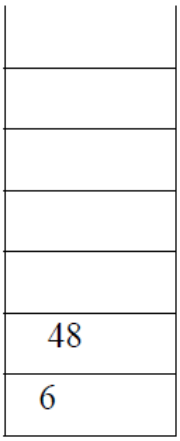
7



8

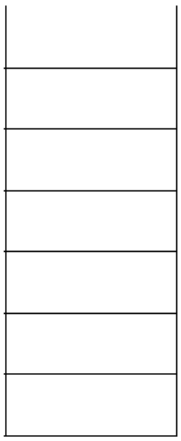


9



10

11



12

يتم التعامل مع تعبير العلامة اللاحقة على انه سلسلة من الرموز, وفي كل مرة يقرأ احد هذه الرموز وذلك من اليسار الى اليمين. اولا يقرأ الرقم 6 ويدفع الى داخل المكدس الفارغ (الارقام (المعاملات) تدفع الى المكدس) بعدها ياتي الرقم 5 فيدفع الى المكدس ايضا, ثم ياتي الرقم 2 ويدفع الى المكدس ايضا حسب القاعدة, الرمز الاخر هو الرقم 3 ويدفع ايضا الى المكدس, بعد دفع الرقم 3 الى المكدس نصل الى

علامة الجمع وكما اسلفنا ان العوامل الرياضية لا تدفع الى المكدس ومايحدث هو ان يتم سحب رقمين من اعلى المكدس واجراء هذه العملية الرياضية عليهما, هنا في اعلى المكدس الرقم 3 وبعدة الرقم 2 يتم سحبهما من المكدس وتجرى عليهم عملية الجمع ليكون الناتج 5 هذا الناتج يعاد دفعة الى المكدس وطبعا سيكون في اعلى المكدس (الخطوة 5), تستمر عملية قراءة رموز تعبير العلامة اللاحقة فنقرأ الرقم 8 وهذا رقم فيدفع الى المكدس كما في الخطوة (6), بعدها نقرأ علامة الضرب وبهذا يتم سحب رقمين من اعلى المكدس وهما الرقم 8 والذي يلية الرقم 5 لتتم عليهم عملية الضرب ويكون الناتج 40 الذي يدفع الى اعلى المكدس (خطوة 7) تستمر عملية قراءة رموز تعبير العلامة اللاحقة حيث علامة الجمع عندها يسحب اعلى رقمين وهم 40 والرقم 5 وتجرى عملية الجمع والناتج هو 45 يدفع الى اعلى المكدس, بعدها يتم قراءة الرقم 3 ويدفع الى اعلى المكدس ثم نقرأ علامة الجمع ونسحب الرقم 45 والرقم 3 من اعلى المكدس وتجرى عليهم عملية الجمع ليكون الناتج 48 يدفع الى اعلى المكدس (الخطوة 10), واخيرا تتم عملية قراءة عامل الضرب ويتم سحب الرقم 48 والرقم 6 لاجراء عملية الضرب وينتج الرقم 288 الذي يدفع الى المكدس وبهذا تنتهي العملية الرياضية نظرا لانتهاء جميع رموز تعبير العلامة اللاحقة, ولم يبقى بالمكدس سوى الناتج الذي هو 288.

العملية تتوقف عندما تنتهي جميع العوامل الرياضية والمعاملات في السلسلة (تعبير العلامة اللاحقة), نتيجة حساب التعبير المتحصلة يتم سحبها من المكدس وستكون هي الوحيدة في المكدس اي معامل واحد يبقى بالمكدس.

مثال 7 : مثال اخر توضيحي لايجاد نتيجة تعبير العلامة اللاحقة باستخدام المكدس:

623+-382/+*2^3+

تكون البداية باستخدام مكدس فارغ.

623+-382/+*2^3+		
الرمز	**المكدس**	**الملاحظات**
6	6	معامل يدفع الى المكدس
2	6, 2	معامل يدفع الى المكدس
3	6, 2, 3	معامل يدفع الى المكدس
+	6, 5	عامل رياضي, يتم سحب اعلى معاملين من المكدس, وتجرى عليهم العملية الرياضية وفقا للعامل الرياضي هنا عملية جمع (2+3) ثم تعاد النتيجة (5) الى اعلى المكدس
-	1	عامل, يسحب اعلى معاملين ويجرى عليهم عملية الطرح (6-5) وتدفع النتيجة (1) الى اعلى المكدس
3	1, 3	معامل يدفع الى المكدس
8	1, 3, 8	معامل يدفع الى المكدس
2	1, 3, 8, 2	معامل يدفع الى المكدس
/	1, 3, 4	عامل القسمة, يسحب اعلى معاملين بالمكدس وتجرى عليهم

		القسمة (8/2) وتدفع النتيجة (4) الى اعلى المكدس. **ملاحظة: المعامل الاول الذي يسحب من المكدس يوضع بعد العامل الرياضي بينما المعامل الثاني يوضع قبل العامل الرياضي**.
+	1, 7	جمع المعاملين (4+3) ودفع النتيجة (7) الى اعلى المكدس
*	7	عملية ضرب لاعلى معاملين بالمكدس (1*7) ودفع النتيجة (7) الى اعلى المكدس
2	7, 2	معامل يدفع الى المكدس
^	49	عامل الرفع (اس) لاعلى معاملين بالمكدس (7^2) وتدفع النتيجة (49) الى اعلى المكدس
3	49, 3	معامل يدفع الى المكدس
+	52	اجراء عملية الجمع على اعلى معاملين بالمكدس (49+3) ودفع النتيجة (52) الى اعلى المكدس, وهي اخر عملية وتمثل النتيجة النهائية.
	52	نتيجة حساب تعبير العلامة اللاحقة

مثال 8: تحويل تعبير العلامة القياسي 1+2*3^4/5-6 الى تعبير العلامة اللاحقة باستخدام المكدس

1+2*3^4/5-6		
سلسلة المخرجات تعبير العلامة اللاحقة	المكدس	رموز تعبير العلامةالقياسية
1		1
1	+	+
12	+	2
12	+*	*
123	+*	3
123	+*^	^
1234	+*^	4
1234	+*^/	/
1234^	+*/	
1234^*	+/	
1234^*5	+/	5
1234^*5	+/-	-
1234^*5/	+-	
1234^*5/+	-	
1234^*5/+6	-	6
1234^*5/+6-	فارغ	تعبير العلامة اللاحقة

مثال 9: حول تعبير العلامة القياسي 4*(2+3)-1 الى تعبير العلامة اللاحقة باستخدام المكدس

4*(2+3)-1		
رموز تعبير العلامة القياسية	**المكدس**	**سلسلة المخرجات تعبير العلامة اللاحقة**
1		1
-	-	1
(	-(	1
2	-(	12
+	-(+	12
3	-(+	123
)	-(	123+
	-	123+
*	-*	123+
4	_*	123+4
	-	123+4*
تعبير العلامة اللاحقة	فارغ	123+4*-

مثال 10: حول تعبير العلامة القياسي ((2+2)*(3-2))/(6-5)*(4+8) الى تعبير العلامة اللاحقة

((2+2)*(3-2))/(6-5)*(4+8)		
عناصر تعبير العلامة القياسي	المكدس	سلسلة المخرجات تعبير العلامة اللاحقة
(	(	
4	(	4
+	(+	4
8	(+	48
)		48+
*	*	48+
(	*(	48+
6	*(	48+6
-	*(-	48+6
5	*(-	48+65
)	*	48+65-
/	*/	48+65-

	/	48+65-*
(	/(	48+65-*
(	/((	48+65-*
3	/((	48+65-*3
-	/((-	48+65-*3
2	/((-	48+65-*32
)	/(	48+65-*32-
*	/(*	48+65-*32-
(	/(*(	48+65-*32-
2	/(*(	48+65-*32-2
+	/(*(+	48+65-*32-2
2	/(*(+	48+65-*32-22
)	/(*	48+65-*32-22+
)	/	48+65-*32-22+*
تعبير العلامة اللاحقة	فارغ	48+65-*32-22+*/

مثال 11 : تحويل تعبير العلامة اللاحقة * – f + e / d – c b a الى تعبير العلامة القياسية

a b – c / d e + f – *		
عناصر تعبير العلامة اللاحقة	المكدس	الملاحظات
a	a	معامل يدفع الى المكدس
b	a,b	معامل يدفع الى المكدس
-		عامل رياضي يتم سحب اعلى عنصرين بالمكدس وهما a, b ووضع العلامة الرياضية بينهما, يدفع الناتج الى المكدس
	(a-b)	تعتبر النتيجة عنصر واحد عند دفعها الى المكدس لذا توضع بين قوسين للتعامل معها على انها عنصر واحد
c	(a-b),c	معامل يدفع الى المكدس
/		عامل رياضي, لذا يسحب اعلى عنصرين بالمكدس وهما (a-b), c وتوضع بينهم العلامة الرياضية لتكون عنصر واحد يدفع للمكدس
	(a-b)/c	يكون العنصر الاول الذي يسحب من المكدس هو العنصر الذي يوضع بعد العلامة الرياضية بينما العنصر الثاني الذي يسحب من المكدس يوضع قبل العلامة الرياضية
d	(a-b)/c,d	معامل يدفع الى المكدس
e	(a-b)/c,d,e	معامل يدفع الى المكدس
+	(a-b)/c	عامل رياضي, يسحب اعلى معاملين بالمكدس وهما d, e

توضع بينهم العلامة الرياضية لتكون عنصر واحد يدفع الى المكدس	(a-b)/c,(d+e)	
معامل يدفع الى المكدس	(a-b)/c,(d+e),f	f
عامل رياضي, يسحب اعلى معاملين في المكدس وهما (d+e), f	(a-b)/c	-
توضع العلامة الرياضية بينهم ليكون عنصر واحد يدفع الى المكدس	(a-b)/c,(d+e)-f	
	(a-b)/c,(d+e)-f	
عامل رياضي يسحب اعلى عنصرين بالمكدس وهما ((a-b)/c), ((d+e)-f) ويوضع بينهم العلامة الرياضية ليتكون عنصر واحد يدفع الى المكدس		*
	(a-b)/c*(d+e)-f	
النتيجة النهائية حيث لايوجد عناصر اخرى اضافية	(a-b)/c*(d+e)-f	

تمرين 2: حول كل تعبير من تعابير العلامة القياسية ادناة الى تعبير العلامة اللاحقة باستخدام المكدس (*الاجابة في نهاية الفصل*)

1. A + B * C – D * E / F
2. A + B ^ C / D * E + F – A
3. A ^ B ^ C / 2 * B + 4
4. A / B + C / D – E * F ^ 2 / B
5. (A + B) * (C – B) / 2 ^ 4
6. (A + B) * (C – D)
7. A – B / (C + D * E)
8. ((A + B) * C – (D – E))/(F + G)

2.7 خوارزمية تحويل تعبير العلامة القياسية الى تعبير العلامة السابقة

يتم فحص تعبير العلامة القياسية من اليمين الى اليسار ويتم تكرار الخطوات التالية على كل عنصر من عناصر التعبير لحين ان يفرغ المكدس

1. قراءة عنصر جديد من تعبير العلامة القياسية.
2. اذا كان العنصر هو معامل فيتم اضافتة الى سلسلة المخرجات.
3. اذا كان العنصر هو عامل رياضي فيتم اجراء مايلي:
   3.1 اذا كان العامل قوس ايمن " (" يتم وضعة في المكدس.
   3.2 اذا كان العامل قوس ايسر ") " في هذه الحالة يتم سحب جميع العوامل الموجودة في المكدس لحين الوصول الى القوس الايمن. جميع هذة العوامل التي سيتم سحبها من المكدس تكتب في سلسلة المخرجات بالتعاقب (اي واحد بعد الاخر).

3.3 اذا كان العنصر عامل رياضي فيتم اجراء مايلي:

3.3.1 اذا كان المكدس فارغ فيتم اضافة العامل الرياضي الى المكدس.

3.3.2 اما اذا كان المكدس غير فارغ, عليك بالقيام بمايلي:

a) ايجاد الافضلية للعامل الرياضي .

b) ايجاد الافضلية للعامل الرياضي الموجود في اعلى المكدس.

c) مقارنة الافضليات للعاملين, فاذا كانت افضلية العامل الرياضي المراد اضافتة اعلى من افضلية العامل الرياضي الموجود في اعلى المكدس في هذة الحالة يتم دفع هذا العامل الى المكدس ليكون في اعلى المكدس. اما اذا كانت افضلية العامل المراد اضافتة اقل من افضلية العامل الرياضي الموجود في اعلى المكدس عندها يتم اخراج العامل الرياضي في اعلى المكدس ويضاف الى سلسلة المخرجات.. تستمر عملية المقارنة مع العامل الذي سيكون في اعلى المكدس بعد سحب العامل السابق وتطبق نفس الطريقة, وهكذا لحين الوصول الى مكدس فارغ او عامل رياضي في اعلى المكدس افضليتة اقل من العامل المراد اضافتة عندها يضاف العامل الى اعلى المكدس.

4. اذا كان هناك عناصر اخرى في تعبير العلامة القياسية, الذهاب الى الخطوة 1. في خلاف ذلك الذهاب الى الخطوة 5.
5. افراغ جميع محتويات المكدس (ازالة كل العوامل من اعلى المكدس) واضافتها الى سلسلة المخرجات.
6. بعد الانتهاء من افراغ المكدس يتم عكس رموز سلسلة المخرجات (اي يكون الرمز الايسر باليمين والرمز الذي باليمين باليسار) حيث ستكون هذه السلسة الناتجة تمثل تعبير العلامة السابقة.

ملاحظة: بالامكان عكس رموز تعبير العلامة القياسية ابتداءا وتطبيق الخوارزمية اعلاه عليها, دون الحاجة الى عكس النتيجة النهائية.

مثال 12: حول تعبير العلامة القياسية **(a – b) / c * (d + e – f / g) الى تعبير العلامة السابقة**

(a – b) / c * (d + e – f / g)		
رموز تعبير العلامة القياسية	**المكدس**	**تعبير العلامة السابقة**
)	)	
g	)	g
/	)/	g
f	)/	gf
-	)/-	gf
	)-	gf/
e	)-	gf/e
+	)-+	gf/e
d	)-+	gf/ed

gf/ed+-		(
gf/ed+-	*	*
gf/ed+-c	*	c
gf/ed+-c	*/	/
gf/ed+-c	*/)	)
gf/ed+-cb	*/)	b
gf/ed+-cb	*/)-	-
gf/ed+-cba	*/)-	a
gf/ed+-cba-	*/	(
gf/ed+-cba-/*	تعكس هذه النتيجة للحصول على تعبير العلامة السابقة	
* / – a b c – + d e / f g	تعبير العلامة السابقة	

مثال 13: حول تعبير العلامة القياسية (1-4)*5+(1-2)/3*2 الى تعبير العلامة السابقة باستخدام المكدس

2*3/(2-1)+5*(4-1)		
رموز تعبير العلامة القياسية	**المكدس**	**سلسلة المخرجات تعبير العلامة السابقة**
)	)	
1	)	1
-	)-	1
4	)-	14
(	فارغ	14-
*	*	14-
5	*	14-5
+	+	14-5*
)	+)	14-5*
1	+)	14-5*1
-	+)-	14-5*1
2	+)-	14-5*12
(	+	14-5*12-
/	+/	14-5*12-
3	+/	14-5*12-3

14-5*12-3	+/*	*
14-5*12-32	+/*	2
14-5*12-32*/+	فارغ	هذة النتيجة يتم عكسها
+/*23-21*5-41	تعبير العلامة السابقة	

مثال 14 : تحويل تعبير العلامة السابقة * / – a b c – + d e f الى تعبير العلامة القياسية
ملاحظة: يتم قراءة عناصر تعبير العلامة السابقة من اليمين الى اليسار

* / – a b c – + d e f		
عناصر تعبير العلامة السابقة	**المكدس**	**الملاحظات**
f	f	معامل يدفع الى المكدس
e	f,e	معامل يدفع الى المكدس
d	f,e,d	معامل يدفع الى المكدس
+	f	عامل رياضي, يتم سحب اعلى معاملين بالمكدس وتوضع العلامة الرياضية بينهما, وهما d, e.. ويتم دفع النتيجة الى اعلى المكدس
	f,(d+e)	**ملاحظة: المعامل الاول الذي يسحب من اعلى المكدس يوضع قبل العلامة الرياضية , بينما المعامل الثاني الذي يسحب من المكدس يوضع بعد العلامة الرياضية.**
-		عامل رياضي
	(d+e)-f	
c	(d+e)-f,c	معامل يدفع الى المكدس
b	(d+e)-f,c,b	معامل يدفع الى المكدس
a	(d+e)-f,c,b,a	معامل يدفع الى المكدس
-	(d+e)-f,c	عامل رياضي, يتم سحب اعلى معاملين بالمكدس وهما a, b.. وضع العلامة الرياضية بينهما ودفع النتيجة الى اعلى المكدس
	(d+e)-f,c,(a-b)	
/	(d+e)-f	
	(d+e)-f,(a-b)/c	
*		
	(a-b)/c * (d+e)-f	تعبير العلامة القياسية

مثال 15 : تحويل تعبير العلامة القياسية (a-b)/c * (d+e)-f الى تعبير العلامة السابقة

(a-b)/c * (d+e)-f		
رموز تعبير العلامة القياسية (من اليمين الى اليسار)	المكدس	سلسلة المخرجات تعبير العلامة السابقة
f		f
-	-	f
)	-)	
e	-)	fe
+	-)+	fe
d	-)+	fed
(	-	fed+
*	-*	fed+
c	-*	fed+c
/	-*/	fed+c
)	-*/)	
b	-*/)	fed+cb
-	-*/)-	fed+cb
a	-*/)-	fed+cba
(	-*/	fed+cba-
	-*	fed+cba-/
	-	fed+cba-/*
	فارغ	fed+cba-/*-
يتم عكس الناتج للحصول على تعبير العلامة السابقة		-*/-abc+def

تمرين 3: احسب نتيجة تعابير العلامة اللاحقة التالية موضحا محتويات المكدس بعد كل عملية (الاجوبة في نهاية الفصل)

1. 5, 3, + , 2 , * , 6 , 9 , 7 , - , / , -
2. 3, 5, + , 6 , 4 , - , * , 4 , 1 , - , 2 , ^ , +
3. 3, 1, + , 2, ^ 7 , 4, - , 2 , * , + , 5 , -
4. 20, 45, + 20 , 10 , - , 15 , + , *

2.8 بعض البرامج اللازمة لتطبيقات المكدس باستخدام لغة C++

2.8.1 اكتب دالة باستخدام C++ لانجاز عمليتي (الدفع والسحب) (push, pop) في مكدس يمثل كمصفوفة.

```
void  PUSH ( int Stack [ ], int &Top, int Val )
{
 if ( Top = = size -1 )
{
   cout << " Stack Overflow " ;
   exit(0) ;     }
else
{
   Top + + ;
   Stack [ Top ] = Val ;
}
} // end PUSH function

void  POP ( int Stack [ ] , int & Top )
{
int R ;
if ( Top = = -1 )
{
   cout<< " Stack Underflow" ;
   exit(0) ;     }
else
{
   R = Stack [ Top ] ;
  Top - - ;
}
}  // end POP function
```

2.8.2 اكمال الدوال اعلاة لتكون صنف كامل

```
class Stack
{
int Data [ 10] ;
int Top ;
public:
Stack ( ) { Top = -1; }
void PUSH ( ) ; // to push an element into the stack .
void POP ( ) ; // to pop an element from the stack.
};
void Stack :: PUSH ( )
{
int Val ;
```

```
cout <<"Enter the value to be push." ;
cin>> Val ;
if ( Top = = 9 )
{
cout << " Stack Overflow";
exit ( 0) ; }
else
{ Top + + ;
Data [ Top ] = Val ;
}
}
void Stack :: POP( )
{
int R ;
if ( Top = = -1 )
{ cout<<"Stack Underflow";
exit (0) ;
}
else
{ R = Data [Top ] ;
Top - - ;
}
}
```

2.8.3 برنامج يستخدم لتحويل تعبير العلامة القياسية الى تعبير العلامة اللاحقة

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>

const int size =50;
char infix[size],postfix[size],stack[size];
int top=-1;
int precedence(char ch);        // دالة للحصول على افضليات العوامل
char pop();                     // دالة لسحب عنصر من المكدس
```

```
char topelement();              // اعادة اعلى عنصر في المكدس
void push(char ch);             // دفع عنصر ما الى المكدس
int main()
{
char ele,elem,st[2];
int prep,pre,popped,j=0,chk=0;
strcpy(postfix," ");
gets(infix);
for(int i=0;infix[i]!=0;i++)
{
if(infix[i]!='('&&infix[i]!=')'&&infix[i]!='^'&&infix[i]!='*'&&infix[i]!='/'&&infix[i]!='+'&&i
nfix[i]!='-')
postfix[j++]=infix[i];
else
 if(infix[i]=='(')
{
elem=infix[i];
push(elem);
}
else if(infix[i]==')')
{
while(popped=pop() != '(')

postfix[j++]=popped;
}
else
{
elem=infix[i];
pre=precedence(elem);    //خزن افضلية العامل الذي ياتي من تعبير العلامة القياسية
ele=topelement();
prep=precedence(ele);    //خزن الافضلية للعامل الموجود في اعلى المكدس

if(pre > prep)

push(elem);
```

```
else
{
while(prep >= pre)

{
if(ele=='#')
break;
popped=pop();
ele=topelement();
postfix[j++]=popped;
prep=precedence(ele);
}
push(elem);
}
}
}

while((popped=pop())!='#')
postfix[j++]=popped;
postfix[j]='\0';

cout<<"\n post fix :"<<postfix<<endl;


system("pause");
return 0;
}

int precedence(char ch)

{
switch(ch)
{
case '^' : return 5;
case '/' : return 4;
```

```
case '*' : return 4;
case '+' : return 3;
case '-' : return 3;
default : return 0;
}
}

char pop()        //دالة لسحب عنصر من المكدس

{
char ret;
if(top!=-1)
{ ret =stack[top];

top--;
return ret;

}
else
return '#';

}

char topelement() // دالة لاعادة العنصر الاعلى في المكدس دون ان تسحبهة من المكدس

{
char ch;

if(top!=-1)
ch=stack[top];
else
ch='#';
return ch;

}
```

```
void push(char ch) // دالة لدفع عنصر الى المكدس
{
if(top!=size-1)
{
top++;
stack[top]= ch;
}
}
```

2.8.4 حساب قيمة تعبير العلامة اللاحقة

```
#include <iostream.h>
#include <stdlib.h>
#include <math.h>
#include <ctype.h>
const int MAX = 50 ;
class postfix
{
        private :

                int stack[MAX] ;
                int top, nn ;
                char *s ;
        public :
                postfix( ) ;
                void setexpr ( char *str ) ;
                void push ( int item ) ;
                int pop( ) ;
                void calculate( ) ;
                void show( ) ;
} ;
postfix :: postfix( )
{
        top = -1 ;
}
void postfix :: setexpr ( char *str )
{
```

```
        s = str ;
}
void postfix :: push ( int item )
{
        if ( top == MAX – 1 )
                cout << endl << “Stack is full” ;
        else
        {
                top++ ;
                stack[top] = item ;
        }
}
int postfix :: pop( )
{
        if ( top == -1 )
        {
                cout << endl << “Stack is empty” ;
                return NULL ;
        }
        int data = stack[top] ;
        top-- ;
        return data ;
}
void postfix :: calculate( )
{
        int n1, n2, n3 ;
        while ( *s )
        {
                if ( *s == ‘ ‘ || *s == ‘\t’ )
                {
                        s++ ;
                        continue ;
                }
                if ( isdigit ( *s ) )
                {
                        nn = *s – ‘0’ ;
                        push ( nn ) ;
                }
                else
                {
                        n1 = pop( ) ;
```

```
                        n2 = pop( ) ;
                        switch ( *s )
                        {
                                case '+' :
                                        n3 = n2 + n1 ;
                                        break ;
                                case '-' :
                                        n3 = n2 – n1 ;
                                        break ;
                                case '/' :
                                        n3 = n2 / n1 ;
                                        break ;
                                case '*' :
                                        n3 = n2 * n1 ;
                                        break ;
                                case '%' :
                                        n3 = n2 % n1 ;
                                        break ;
                                case '$' :
                                        n3 = pow ( n2 , n1 ) ;
                                        break ;
                                default :
                                        cout << "Unknown operator" ;
                                        exit ( 1 ) ;
                        }

                        push ( n3 ) ;
                }
                s++ ;
        }
}
void postfix :: show( )
{
        nn = pop ( ) ;
        cout << "Result is: " << nn ;
}

void main( )
{
        char expr[MAX] ;
        cout << "\nEnter postfix expression to be evaluated : " ;
```

```
        cin.getline ( expr, MAX ) ;
        postfix q ;
        q.setexpr ( expr ) ;
        q.calculate( ) ;
        q.show( ) ;
}
```

تمرين 4: حول تعابير العلامة القياسية التالية الى مايكافئها من تعابير العلامة اللاحقة موضحها حالة المكدس:

1. ((A – B) * (D / E)) / (F * G * H)
2. (A + B ^ D) / (E – F) + G
3. A * (B + D) / E – F – (G + H / K)
4. A * (B + (C + D) * (E + F) / G) * H
5. A + ((B + C) + (D + E) * F) / G
6. NOT A OR NOT B AND NOT C
7. NOT (A OR B) AND C

تمرين 5: اكتب تعبير العلامة القياسية المكافيء لتعابير العلامة اللاحقة التالية:

1. 10 , 3 , * , 7 , 1 , - , * , 23 , +
2. 12, 7, 3, - , / , 2 , 1 , 5 + , * , +
3. ABCD^*E/+F-

2.9 اجوبة التمارين:

تمرين 1:

1. a b + c d - *
2. a b c d e * + / -
3. a b + c * d e - - f g + /

تمرين 2:

9. A B C * + D E * F / –
10. A B C ^ D / E * + F + A –
11. A B ^ ^ C 2 / B * 4 +
12. A B / C D / + E F 2 ^ * B / –
13. A B + C B - * 2 4 ^ /

14. A B + C D – *
15. A B C D E * + / –
16. A B + C * D E – – F G + /

تمرين :3

1. 13
2. 25
3. 17
4. 1625

تمرين 4:

1. AB – DE / * FG * H * /
2. ABD ^ + E F - / G +
3. A B D + * E / F – G H K / + -
4. A B C D + E F + * G / + * H *
5. A B C + D E + F * + G / +
6. A NOT B NOT C NOT AND OR
7. A B OR NOT C AND

تمرين :5

1. 10 * 3 * (7 – 1) + 23
2. 12 / (7 – 3) + (1 + 5) * 2
3. A+B*C^D/E-F