

مقرر الخوارزميات و بنى المعطيات 1

جلسة العملي الثالثة

(الفصل الثاني 2023-2024)

تطبيقات على المكس

تمرين 1: أوجد تحويل infix المقابل لتعبير RPN التالي:
 $324+*95-8/-$

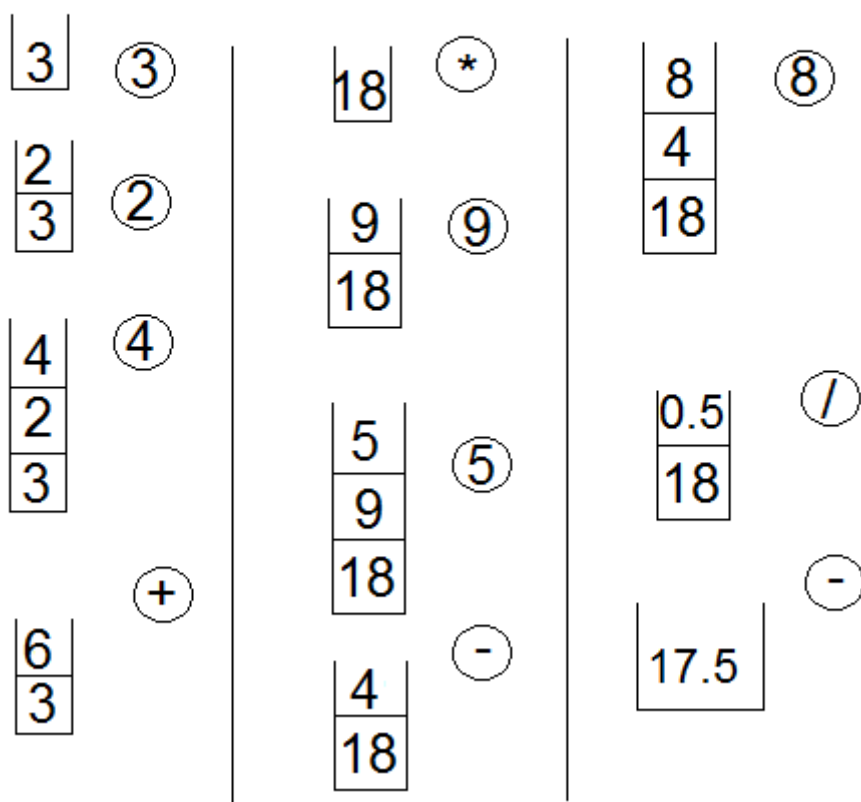
الحل:

نمسح السلسلة من اليسار حتى الوصول إلى معامل عندها نقوم بتوسيط
المعامل لآخر قيمتين قبله بالتالي يكون تعبير infix الموافق هو

$$\begin{aligned} & 324+*95-8/- \\ & 3(2+4)*95-8/- \\ & 3*(2+4) \underline{95-8}/- \\ & 3*(2+4)(9-5) \underline{8}/- \\ & 3*(2+4)(9-5)/ \underline{8}- \\ & (3*(2+4))-((9-5)/ 8) \end{aligned}$$

تمرين 2: احسب قيمة التعبير RPN التالي اعتمادا على طريقة المكس
 $324+*95-8/-$

$$324+*95-8/-$$



تمرين 3: حول التعبير $((9-5)/8)-(3*(2+4))$ إلى صيغة RPN وذلك :
1- باستخدام طريقة الأشجار.



جامعة
المنارة

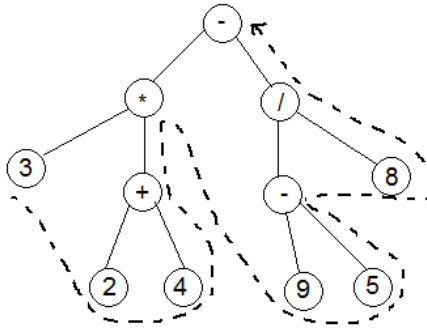
2- باستخدام طريقة الانتقال والحذف:

3- باستخدام طريقة المكس

الحل:

1- طريقة الأشجار

$$3*(2+4)-((9-5)/8)$$



فيكون الناتج :

$$324+*95-8/-$$

2- طريقة الانتقال والحذف

$$(3*(2+4))-((9-5)/8))$$

$$((3*(2+4))-((9-5)/8))$$

فيكون الناتج :

$$324+*95-8/-$$

3- باستخدام طريقة المكس

$$3*(2+4)-(9-5)/8$$

324	$\begin{array}{ c } \hline + \\ \hline (\\ \hline * \\ \hline \end{array}$	④	324+*9	$\begin{array}{ c } \hline (\\ \hline - \\ \hline \end{array}$	⑨	324+*95-	$\begin{array}{ c } \hline - \\ \hline \end{array}$	①
324+	$\begin{array}{ c } \hline * \\ \hline \end{array}$	①	324+*9	$\begin{array}{ c } \hline - \\ \hline (\\ \hline - \\ \hline \end{array}$	①	324+*95-	$\begin{array}{ c } \hline / \\ \hline - \\ \hline \end{array}$	①
324+*	$\begin{array}{ c } \hline - \\ \hline \end{array}$	①				324+*95-8	$\begin{array}{ c } \hline / \\ \hline - \\ \hline \end{array}$	⑧
324+*	$\begin{array}{ c } \hline (\\ \hline - \\ \hline \end{array}$	①	324+*95	$\begin{array}{ c } \hline - \\ \hline (\\ \hline - \\ \hline \end{array}$	⑤	324+*95-8/	$\begin{array}{ c } \hline - \\ \hline \end{array}$	
						324+*95-8/-	$\begin{array}{ c } \hline \\ \hline \end{array}$	

تمرين 4: اكتب برنامج لتحويل العبارة المدخلة من قبل المستخدم إلى صيغة الترميز البولوني العكسي.

الحل:

إن خوارزمية التحويل من الترميز infix إلى الترميز RPN باستخدام المكس هي كما يلي:

1. تهيئة مكس فارغ للمعاملات.

2. طالما أنه ليس هناك خطأ ولم نصل إلى نهاية التعبير الحسابي نقوم بما يلي:

a. جلب الرمز التالي (ثابت , متحول , معامل , قوس يميني , قوس يساري).
b. إذا كان الرمز:

i. قوس يساري: قم بدفعه إلى المكس.
ii. قوس يميني: قم بإخراج القيم من المكس و قم بعرضها على الشاشة لحين الوصول إلى قوس يساري فلا تقم بعرضه.

iii. معامل: إذا كان المكس فارغاً أو الرمز ذو أولوية أعلى من القيمة الموجودة في قمة المكس , قم بدفع الرمز في المكس.

وإلا أخرج القيمة الموجودة في أعلى المكس و اعرضها (ملاحظة القوس اليساري في المكس يملك أولوية دنيا).

iv. قيمة: قم بعرضها.

3. عند الوصول إلى نهاية التعبير قم بإخراج و عرض عناصر المكس إلى أن يصبح فارغاً.

```
#include "stdafx.h"
```

```
#include<string>
```

```
#include<cassert>
```

```
#include "Stack.h"
```

```
using namespace std;
```

```
string RPN(string exp);
```

```
int main()
```

```
{
```

```
    string exp;
```

```
cout<<"NOTE:Enter # for infix expression to  
stop\n";
```

```
for (;;)
{
    cout<<"\nInfix Expression ?";
    cin>>exp;
    if (exp=="#") break;
    cout<<"RPN Expression is  
"<<RPN(exp)<<endl;
}
```

```
system("pause");
return 0;
```

```
}
```

```
string RPN(string exp)
```

```
{
    char token , topToken ;
    Stack opStack;
    string RPNexp;
    const string BLANK = " ";
    RPNexp = "" ;
    for (int i=0;i<exp.length();i++)
    {
        token=exp[i];
        switch(token)
        {
            case ' ':break;
```

```
case '(':opStack.push(token); break;
case ')':for( ; ; )
    { assert(!opStack.empty());
      topToken=opStack.top();
      opStack.pop() ;
      if (topToken== '(') break;
      RPNexp.append(BLANK+topToken );
    } break;
case '+':case '-':case '*':case '/':for( ; ; )
    { if (opStack.empty()
      ||opStack.top()=='(' ||
      (token=='*' || token=='/') &&
      (opStack.top()=='+' ||
      opStack.top()=='-'))
        { opStack.push(token);break;
        }
      else {
        topToken=opStack.top();
        opStack.pop();
        RPNexp.append(BLANK
        +topToken);
      }
    }
    break;
```




جامعة
المنارة

```
default:RPNexp.append(BLANK+token);
    }
}
for ( ; ; )
{
    if (opStack.empty()) break;
    topToken=opStack.top();
    opStack.pop();
    if (topToken!='(')
        RPNexp.append(BLANK+topToken);
else
    {
        cout<<"**** ERROR IN INFIX EXPRESSION
        *** \n";
        break;
    }
}
return RPNexp;
}
```

تمرين 5: اكتب برنامج يقوم بالتحويل من النظام العشري إلى الثنائي

```
#include "stdafx.h"
#include<iostream>
#include "stack.h"
using namespace std;
int main( )
```

```
}unsigned number,remainder;
Stack stackofRemainders;
char response;
do
}
cout<<"Enter positive integer to convert :";
cin>>number;
while ( number!=0)
{ remainder=number%2;
stackofRemainders.push(remainder);
number/=2;
}
cout<<"Base-two representation :";
while (!stackofRemainders.empty() )
{ remainder=stackofRemainders.top() ;
stackofRemainders.pop() ;
cout<<remainder;
}
cout<<endl;
cout<<"\nMore (Y or N)?";
```

```
cin>>response؛
```

```
{while (response=='Y' || response=='y');
```

```
return 0؛
```

```
}
```

وظيفة:

- 1- اكتب برنامج يقوم بالتحويل من النظام العشري إلى النظام الست عشري
- 2- حول التعبير $(3+6)/(5-2) - (2+5)*7$ إلى صيغة RPN وذلك :
 - 1- باستخدام طريقة الأشجار.
 - 2- باستخدام طريقة الانتقال والحذف.
 - 3- باستخدام طريقة المكس