

Stack

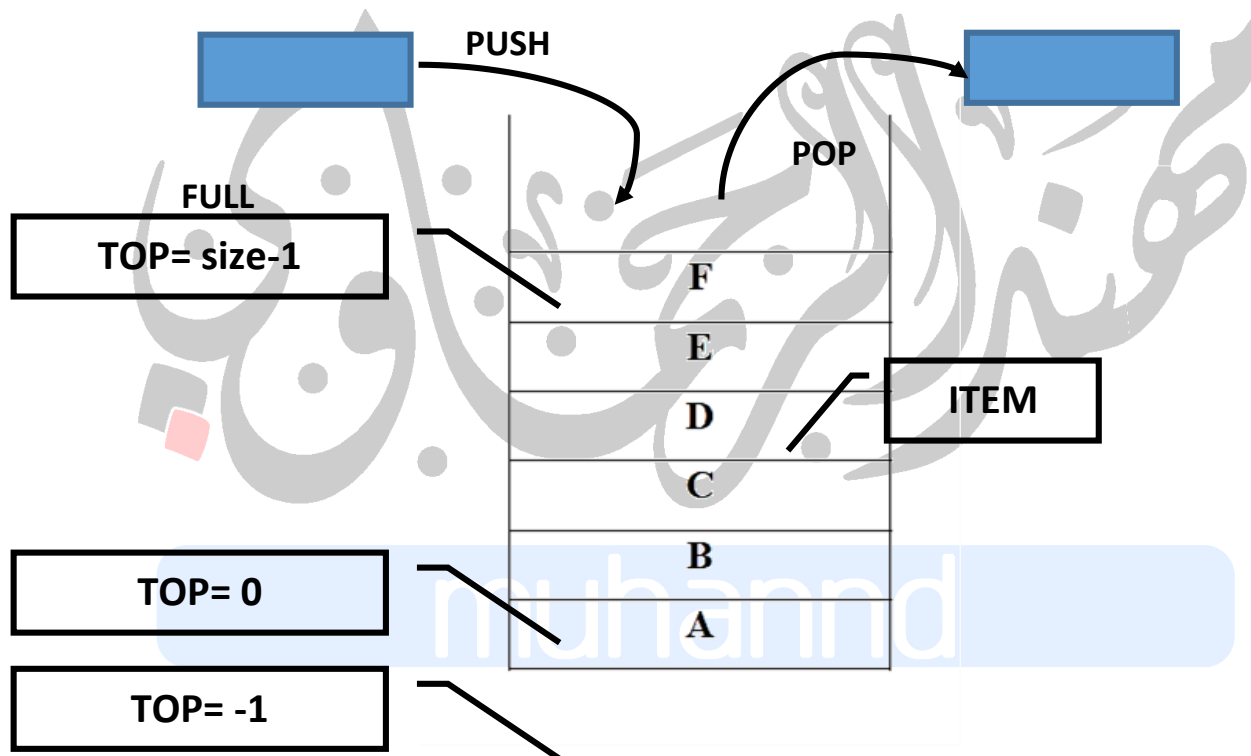
المكدس

هو هيكل بياني خطي يتكون من نهايتين نهاية مفتوحة تسمى قمة المكدس top of stack ونهاية ثانية مغلقة وتجرى عمليات الإضافة والحذف والوصول الى أي عنصر من خلال النهاية المفتوحة او عن طريق قمة المكدس وبذلك يطلق عليه مصطلح (Last In First Out) LIFO ويعني من يدخل اخر يحصل على الخدمة أولاً وتسمى عملية الإضافة PUSH والحذف POP وان المتغيرات المستخدمة هي

TOP: متغير صحيح يمثل قمة المكدس

ITEM: ويمثل العنصر المراد اضافته

STACK: مصفوفة أحادية لتمثيل عنصر المكدس



ويتم الإعلان عن المكدس بالشكل التالي:

```
data-type Stack-name [ size ];
```

Data type : النوع البياني للمكدس.

Stack-name : اسم المكدس ويراعى فيه شروط تسمية المتغيرات.

Size : ويمثل حجم المكدس (أي عدد عناصر المكدس).

```
Int stackage [10];
```

```
Int stacknum [30];
```

```
Float stackdegree [5];
```

ملاحظات

- الوصول الى عنصر يتم عبر TOP . لا يمكن الوصول الي اي عنصر الا بعد حذف جميع العناصر الاعلى منه.
- تتطلب المكدة تعريف مؤشر واحد يشير الي قمة المكدة نطلق عليه Top
- تزداد قيمة المؤشر بمقدار واحد عند كل عملية اضافة وتنقص بمقدار واحد عند كل عملية حذف
- يصبح المكس خالياً اذا كان المؤشر يساوي - 1 أي $top = -1$.
- تستخدم الحلقة التكرارية For دائما مع المكس عن عملية القراءة والطباعة والمعالجة

دوال خاصة بعملية القراءة والطباعة وهي كالتالي:

❖ برنامج فرعي لقراءة عناصر المكس

```
void push(int stack[size],int& top,int item)
{
if(top==size-1)
cout<<"ERROR... THE STACK ISFULL!!"<<endl;
else
{
top++;
stack[top]=item;
}
}
```

❖ برنامج فرعي لحذف عناصر المكس

```
void pop(int stack[size],int& top,int& item)
{
if(top==-1)
cout<<"ERROR... THE STACK IS EMPTY!!"<<endl;
else
{
item=stack[top];
top--;
}
}
```

❖ برنامج فرعي لطباعة عناصر المكس

```
void print (int stack[size],int& top)
{
While(top>-1)
pop(stack,top,item);
cout<<item<<" ";
Top--;}
}
```

اكتب برنامج يقوم بإضافة ٥ عناصر صحيحة الى مكس ثم يقوم بحذف اخر ثلاثة منها وطباعة النتائج؟

```

#include<iostream.h>
const int size=10;
void push(int stack[size],int& top,int item)
{
    if(top==size-1)
        cout<<"ERROR... THE STACK ISFULL!!"<<endl;
        else
            {
                top++;
                stack[top]=item;
            } }
void pop(int stack[size],int& top,int& item)
{
    if(top==-1)
        cout<<"ERROR... THE STACK IS EMPTY!!"<<endl;
        else
            {
                item=stack[top];
                top--;
            } }
main()
{
    int stack[size],top,item,i;
    for(i=0;i<size;i++)
    {
        cout<<"enter item"<<endl;
        cin>>item;
        push(stack,top,item);
    }
    for(i=0;i<3;i++)
    {
        pop(stack,top,item);
        cout<<item<<" ";
    }
}

```

اكتب برنامج بلغة C++ يقوم بإدخال درجات عشرة طلاب في مادة الهياكل ثم يقوم بطباعة كلمة successful إذا كان ناجح وكلمة un successful إذا كان راسب

```

#include<iostream.h>
const int size=10;
void push(int stack[size],int& top,int item)
{
    if(top==size-1)
        cout<<"ERROR... THE STACK ISFULL!!"<<endl;
        else
            {
                top++;
                stack[top]=item;
            } }
void pop(int stack[size],int& top,int& item)
{
    if(top==-1)
        cout<<"ERROR... THE STACK IS EMPTY!!"<<endl;
        else
            {
                item=stack[top];
                top--;
            } }
main()
{
    int stack[size],top,item,i;
    for(i=0;i<size;i++)
    {
        cout<<"enter item"<<endl;
        cin>>item;
        push(stack,top,item);
    }
    for(i=0;i<size;i++)
    {
        pop(stack,top,item);
        if(item>=50)
            cout<< "successful"<<item<<endl;
        else
            cout<<"un successful "<<item<<endl;
    }
}

```

اكتب برنامج يقوم بقراءة مكدسين ثم قم بأجراء التالي :

- ١- انشاء مكس ثالث يحتوي على حاصل جمع الاعداد (اذا كان العدد في المكس الأول زوجي والذي يقابله في الثاني زوجي)
٢- انشاء مكس رابع يحتوي على القيمة الضعف للعناصر في المكس الثالث

```
#include<iostream.h>
```

```
const int size=10;
```

```
void push(int stack[size],int& top,int item)
```

```
{
  if(top==size-1)
    cout<<"ERROR... THE STACK ISFULL!!"<<endl;
  else
  {
    top++;
    stack[top]=item;
  } }
```

```
void pop(int stack[size],int& top,int& item)
```

```
{
  if(top==-1)
    cout<<"ERROR... THE STACK IS EMPTY!!"<<endl;
  else
  {
    item=stack[top];
    top--;
  } }
```

```
void print (int stack[size],int& top)
```

```
{
  While(top>-1)
    pop(stack,top,item);
  cout<<item<<" ";
  Top--;
}
```

```
main()
```

```
{
  int stack1[size],top1,item1,i;
  int stack2[size],top2,item2,i;
  int stack3[size],top3,item3,i;
  int stack4[size],top4,item4,i;
```

```
for(i=0;i<size;i++)
```

```
{
```

```
cout<<"enter item"<<endl;
cin>>item1;
push(stack1,top1,item1);
}
```

```
for(i=0;i<size;i++)
{
cout<<"enter item"<<endl;
cin>>item2;
push(stack2,top2,item2);
}
```

```
for(i=0;i<size;i++)
{
pop(stack1,top1,item1);
if(item1%2==0){
pop(stack2,top2,item2);
if(item1%2==0)
item3=item1+item2;
cout<<item3;
push(stack3,top3,item3);
}}
```

```
for(i=0;i<size;i++)
{
pop(stack3,top3,item3);
item4=item3*item3;
push(stack4,top4,item4);
}
print(stack4,top4,item4);
}
```

اكتب برنامج بلغة C++ يقوم بتحويل التعبير من صيغة Postfix الى صيغة Infix باستخدام مكسدين ؟

```
#include<iostream.h>
const int size=100;
void push(int stack[size],int& top, int item)
{
  if(top==size-1)
    cout<<"ERROR... THE STACK ISFULL!!"<<endl;
    else
    {
      top++;
      stack[top]=item;
    }
}
```

```
void pop(int stack[size],int& top,int& item)
{
  if(top==--1)
    cout<<"ERROR... THE STACK IS EMPTY!!"<<endl;
    else
    {
      item=stack[top];
      top--;
    }
}
```

```
void print (int stack[size],int& top)
{
  While(top>-1)
  pop(stack,top,item);
  cout<<item<<" ";
  Top--;
}
```

```
String infixToPostfix (string arg, string stack1 [size], int &top1, string stack2 [size],
int&top2)
{
String postfix;

  For(int i=0;i<arg.length;i++)
  {
    If (arg[i]=='(' )
    Push(stack1,top,arg[i]);
    Else If (arg[i]=='(') )
```

```

{
    While(stack[top]!='')
    {
        Postfix=postfix+stack[top];
        Pop(stack2,top,item);
    }
}
Else If (arg[i]>=0&&arg[i]<=9)
Push(stack2,top,arg[i]);
Else If (arg[i]>='a'&&arg[i]<='z')
Push(stack2,top,arg[i]);
Else If (arg[i]>='A'&&arg[i]<='Z')
Push(stack2,top,arg[i]);
Else if (arg[i]=='*' || arg[i]=='/' || arg[i]=='+' || arg[i]=='-' || arg[i]=='%')
Push(stack1,top,arg[i]);
}
}

main()
{
int stack1[size],top1,item1,i;

int stack2[size],top2,item2

string exp;

cout<<"enter Expression "<<endl;
cin>>exp;

infixToPostfix (arg, stack1, top1, stack2, top2);

print (stack1, top1);

}

```