



المكدس STACK

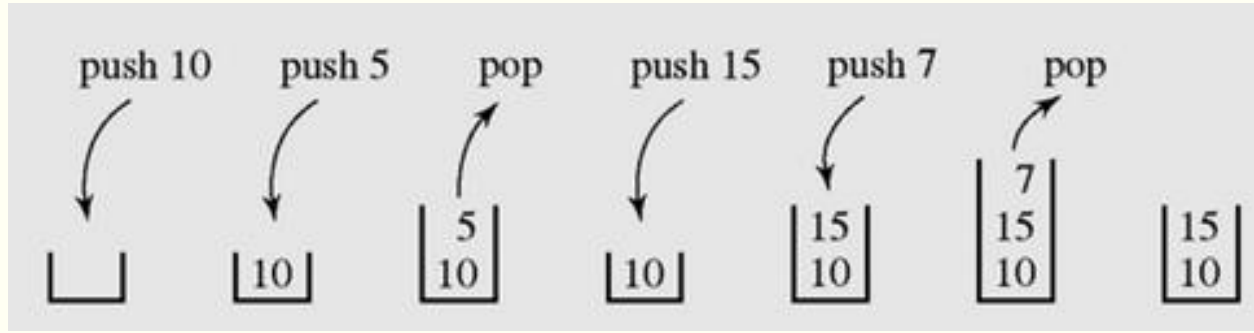
المحاضرة السابعة
م. لمى السبع

المكدس (Stack)

- هي عبارة عن خط انتظار لمجموعة من البيانات، ما يميز هذا الخط بأنه مفتوح من اتجاه واحد فقط.
- البيانات تدخل وتخرج من بوابة واحدة.
- يطلق على stack بـ LIFO وهذا الرمز يعني بأن آخر عنصر دخل إلى المكدس هو أول عنصر يغادره، وهذا بديهي بما أنه لا يوجد لدينا إلا فتحة واحدة لإدخال وإخراج العناصر.
- يمكننا أن نشبهها بحاملة الأقراص المدمجة فآخر قرص تقوم بوضعه هو أول قرص تقوم بإخراجه، أو كمجموعة كتب متراسة فوق بعضها البعض.

العمليات على المكس

- إنشاء مكس جاهز للتهيئة. `emptyStack`
- إضافة عنصر الى المكس. `Push`
- حذف عنصر من المكس `Pop`
- قراءة العنصر الموجود في قمة المكس. `Top`
- اختبار كون المكس فارغا. `isEmpty`
- تفريغ المكس بشكل كامل. `cleanup`
- قراءة قيمة من أعلى المكس دون حذفها `peek`



```
#include <iostream.h>
const lmax=100;

struct stack{
int top;
int val[lmax];
};
//-----
void emtyStack(stack &s)
{s.top=-1;}
//-----
char isEmpty(stack s)
{
if (s.top==-1)
return 'y';
else
return 'n';
}
```

```
void push(stack &s,int e)
{
if (s.top==lmax-1)
cout <<"error:s= full.\n";
else{
++s.top;
s.val[s.top]=e;
}
}
//-----
void pop (stack &s)
{
if (isEmpty(s)=='y')
    cout <<"error:s= empty.\n";
else
--s.top;
}
```

```
void cleanUp (stack s){  
  
    if (isEmpty(s)=='y')  
  
        cout <<"stack clean already";  
  
    else  
  
        emptyStack(s);  
  
}  
  
//.....  
  
int top(stack s){  
  
    if (isEmpty(s)=='y')  
  
    else    cout <<"error:s= empty.\n";  
  
    return s.val[s.top];  
  
}
```

تمرين أول

- اكتب خوارزمية تقوم بالتحقق من صحة علاقة رياضية حيث يكون عدد اقواس الفتح مساوي لعدد اقواس الاغلاق. وبتسلسل صحيح باستخدام مفهوم المكدرات.
- فكرة الحل.....
- انظر الملحق من أجل الكود وسيتم تطبيقه في جلسة العملي.

تمرين ثاني

- معتمدا مفهوم المكدرات ، حيث يمكن للتعبير (prefix) اكتب خوارزمية لحساب قيمة تعبير رياضي تم ادخاله بالشكل المصدر
- ان يحوي ارقاما والعمليات الثنائية البسيطة: + ، - ، * ، / (دون اقواس)