

مقرر برمجة ٢ الجلسة الثالثة عملي

اكتب برنامج يقوم بالاستفادة من stack ومبدأ LIFO في عملية تحويل العدد من النظام العشري إلى النظام الثنائي:

```
#include<iostream>
#include<stdlib.h>
using namespace std;

const int stacksize=50;
int*p ; int stack[stacksize];

void push (int i){ if (p==(stack+stacksize))
{ cout<<"stack over flow"; exit(1); }
*p=i; p++;
}

void pop(int &i) {if(p==stack)
{cout<<"stack under flow";exit(1);}
p--;i=*p;
}
```

```
void main(){
int x;int i,n=0;
while(1)
{ cout <<"enter positive number:"; cin >>x;
cout<<endl;
if (x<0) break;
p=stack;
if (x==0)cout<<"0";
while (x!=0)
{ i=x%2; push(i); n++;
x=x/2;}
while (n>0)
{ pop(i); cout<<i<<" ";n--;}
cout<<endl; }
system("pause");}
```

```
#include<iostream.h>
using namespace std;
struct Date
{   int day;           //1-31
    int month;        //1-12
    int year;         //any valid year
};
Void main(){
Date independence={17,4,1946};

cout<< independence.day<<" " <<
independence.month<<endl;
system("pause");
}
```

أكتب برنامجاً عاماً يتضمن: تعرف النمط Date الذي يستخدم للتعامل مع بيانات التاريخ (اليوم، الشهر، السنة):
وصرح عن متحول من هذا النمط واطبعه

```
void main() {  
    // We can add variables by separating them  
    with a comma here  
    // Put data into the first structure  
    myCar myCar1, myCar2;  
    myCar1.brand = "BMW";  
    myCar1.model = "X5";  
    myCar1.year = 1999;  
    // Put data into the second structure  
    myCar2.brand = "Ford";  
    myCar2.model = "Mustang";  
    myCar2.year = 1969;  
  
    // Print the structure members  
    cout << myCar1.brand << " " <<  
    myCar1.model << " " << myCar1.year << "\n";  
    cout << myCar2.brand << " " <<  
    myCar2.model << " " << myCar2.year << "\n";  
    system("pause");  
}
```

أكتب برنامجاً عاماً يتضمن :
عرف النمط myCar الذي يستخدم للتعامل مع بيانات شركة سيارات (الشركة،
الموديل، السنة) وصرح عن متحولين من هذا النمط واطبعهما

```
#include <iostream>  
#include <string>  
using namespace std;  
struct myCar {  
    string brand;  
    string model;  
    int year;  
};
```

```
#include <iostream>
using namespace std;
struct Point {
    float x_coordinate;
    float y_coordinate;
};
struct Circle {
    Point center;
    float radius;
};
void input_point(Point &po) {
    cout << "Enter x_coordinate: ";
    cin >> po.x_coordinate;
    cout << "Enter y_coordinate: ";
    cin >> po.y_coordinate;
}
void output_point(Point po) {
    cout << "P( " << po.x_coordinate << " , " <<
po.y_coordinate << " ) ";
}
```

أكتب برنامجاً عاماً يتضمن :

تعريف سجل اسمه **point** يعبر عن نقطة بحيث تكون حقول معطياته:
الاحداثيات على المحور الأفقي **x_cor** و الاحداثيات على المحور الشاقولي
y_cor
أما العمليات فهي:

١- تابع **input_point** لادخال البيانات عن طريق تمرير وسائط
٢- تابع **output_point** لطباعة الاحداثيات
تعريف سجل اخر اسمه **circle** يعتبر دائرة بحيث تكون حقول معطياته :
أحداثيات المركز **center** ونصف القطر **radius**.
أما العمليات فهي:

١ . تابع **input_circle** لادخال البيانات
٢ . تابع **output_circle** لطباعة البيانات
٣ . تابع **calaulate_area** يعيد مساحة دائرة .
٤ . استخدم التوابع السابقة في برنامج رئيسي لتنفيذ مايلي: ادخال بيانات الدائرتين
التاليتين
C1((0,0,10,5),c2((-5,6),5)
طباعة بيانات الدائرتين ومساحتهما

```
Circle input_circle() {
    Circle ci;
    cout << "Enter center coordinates...\n";
    input_point(ci.center);
    cout << "Enter radius: ";
    cin >> ci.radius;
    return ci;
}
void output_circle(Circle ci) {
    output_point(ci.center);
    cout << " - Radius = " << ci.radius << endl;
}
double calculate_area(Circle ci) {
    return 3.14 * ci.radius * ci.radius;
}
Circle compare_circle(Circle ci1, Circle ci2) {
    return (calculate_area(ci1) >=
calculate_area(ci2)) ? ci1 : ci2;
}
```

```
void main() {
    cout << "Enter the data of circle#1...\n";
    Circle c1 = input_circle();
    cout << "-----" << endl;
    cout << "Enter the data of circle#2...\n";
    Circle c2 = input_circle();
    cout << "-----" << endl;
    cout << "Printing the data of circle#1...\n";
    output_circle(c1);
    cout << "-----" << endl;
    cout << "Printing the data of circle#2...\n";
    output_circle(c2);
    cout << "-----" << endl;
    cout << "Finding the circle with the highest
area...\n";

    cout << "Area = " << calculate_area(c1) << endl;
    cout << "-----" << endl;
    system("pause");}
```

انتهت الجلسة