

```
def appreciations(avareg):
    if avareg >= 85 and avareg <= 100:
        return "Excellent"
    elif avareg >= 75 and avareg <= 85:
        return "vere good"
    elif avareg >= 65 and avareg <= 75:
        return "good"
    elif avareg >= 50 and avareg <= 65:
        return "passable"
    else:
        return "residuuum"

def addstudent(student1):
    x = 0
    degree = []
    while x == 0:
        na = str(input("enter name student: "))
        nu = int(input("enter count number degree: "))
        sum = 0
        n = nu
        while n > 0:
            x=int(input())
            sum += x
            n -= 1
            degree.append(x)

        student = {
            "name": na,
```

```

        "degre": degre,
        "sum": sum,
        "avareg": sum/nu,
        "appreciation": appreciations(sum/nu)
    }
    student1.append(student)
    x = int(input("do you want to continu enter 0: "))
def printstudent(student1):
    serch = str(input("enter one student to sow data or enter all to swo all data to student: "))
    if serch == "one":
        ser = str(input("enter name student: "))
        for stu in student1:
            if serch == "one":
                if stu["name"]==ser:
                    print(f"name = {stu['name'].center(8)} sum = {str(stu['sum']).center(8)} avareg = {str(stu['avareg']).center(8)} appreciation = {stu['appreciation'].center(10)}")
                    break
                else:
                    continue
            elif serch == "all":
                print(f"name = {stu['name'].center(8)} sum = {str(stu['sum']).center(8)} avareg = {str(stu['avareg']).center(8)} appreciation = {stu['appreciation'].center(10)}")

student1 = []
addstudent(student1)
printstudent(student1)

```

```
class node:
    def __init__(self, data=None, next=None, seco=None):
        self.data=data
        self.next=next
        self.seco=seco
```

```
class linked:
    def __init__(self):
        self.head=None
        self.after=None
    def add_item(self, val):
        if self.head is None:
            self.head=node(val)
            self.after=self.head
        else:
            while self.head.next:
                self.head=self.head.next
            self.head.next=node(val)
            self.head.next.seco=self.head
            self.head=self.head.next
            self.head=self.after
    def delete(self):
        if self.head.next is None:
            self.head=None
        else:
            self.head=self.after
            self.after=self.head.next
```

```

        self.head=self.head.next
        self.head.seco.next=None
        self.head.seco=None
def show(self):
    if self.head is None:
        print("*" * 40)
        print("sorry not found elements.")
        print("*" * 40)
    else:
        while self.head is not None:
            print(self.head.data)
            self.head=self.head.next
        self.head=self.after

```

```

l=linked()
condition=1
while condition:
    print("enter 1 to add item")
    print("enter 2 to delete item")
    print("enter 3 to show value for item")
    print("enter 4 to close programmer")
    con=int(input("enter number to funcation: "))
    if con==1:
        val=int(input("enter value to element: "))
        l.add_item(val)
    elif con==2:
        l.delete()

```

```
        l.delete()
    elif con==3:
        l.show()
    elif con==4:
        condition=0
        print("exit to programmer.")
        break;
```

=====

```
def primary(*value):
    for val in value:
        count = 0
        result = val
        while result > 0:
            if val % result == 0:
                count += 1
                result -= 1
        if count == 2:
            print(val)
```

```
def evennumber(*value):
    for val in value:
        if val%2==0:
            print(val)
```

```
def oddnumber(*value):
```

```
for val in value:
    if val%2!=0:
        print(val)
```

```
print("enter primary to find numbers primary")
print("enter even to find numbers even")
print("enter odd to find numbers odd")
valu=str(input())
num=int(input("enter number of numbers: "))
list1=[]
print("enter number")
while num>0:
    list1.append(int(input()))
    num -= 1
if valu=="primary":
    primary(*list1)
elif valu=="odd":
    oddnumber(*list1)
elif valu=="even":
    evennumber(*list1)
```

=====

```
def plurals(*value):
    plu = 0
    for val in value:
        plu += val
    print("sum of numbers is:", plu)
```

```
    plu += val
    print(f"the plural is: {plu}")
```

```
def supstruction(*value):
    sup = 0
    for val in value:
        sup = val - sup
    print(f"the supstruction is: {sup * -1}")
```

```
def multiply(*value):
    mul = 1
    for val in value:
        mul *= val
    print(f"the multiply is: {mul}")
```

```
def department(value1,value2):
    print(f"the departement is: {value1/value2}")
```

```
def roots(value):
    x = 1
    y = 0
    while x < value / 2:
        if x * x == value:
            print(f"the number {x} is root number {value}")
            y = 1
            break
    else:
```

```
        y = 0
    x += 1
    if y == 0:
        print(f"the number {value} is not root")
```

```
def absolute(*value):
    for val in value:
        if val < 0:
            print(f"the absolute valeu of number: {val} is: {val * -1}")
        else:
            print(f"the absolute valeu of number: {val} is: {val}")
```

```
def Power(value1,value2):
    po = 1
    while value2 > 0:
        po *= value1
        value2 -= 1
    print(f"the result is: {po}")
```

```
def Total(*value):
    for val in value:
        tot = 0
        sho = val
        while val > 0:
            tot += val
            val -= 1
        print(f"the total number {sho} is: {tot}")
```



```
def Milled(*value):
    for val in value:
        tot = 1
        sho = val
        while val > 0:
            tot *= val
            val -= 1
        print(f"the milled number {sho} is: {tot}")
```

```
condition = 1
while condition != 0:
    x=int(input())
    operation = str(input())

    if operation == "+":
        y = int(input())
        plurals(x,y)
    elif operation == "-":
        y = int(input())
        supstruction(x,y)
    elif operation == "*":
        y = int(input())
        multiply(x,y)
    elif operation == "/":
        y = int(input())
        department(x,y)
    elif operation == "root":
```

```

        roots(x)
    elif operation == "apsolute":
        apsolute(x)
    elif operation == "^":
        y = int(input())
        Power(x, y)
    elif operation == "total":
        Total(x, 4, 8, 6)
    elif operation == "milled":
        Milled(x, 2, 4)

tests = str(input("do you want to continue? enter yes or enter no: "))
if tests == "yes":
    condition = 1
else:
    condition = 0

```

=====

```

class node:
    def __init__(self,data=None,next=None,seco=None,befor=None,after=None):
        self.data=data
        self.next=next
        self.seco=seco
        self.befor=befor
        self.after=after
class linked:
    def __init__(self):

```

```

def __init__(self):
    self.head=None
    self.last=None
def add_row(self,val):
    if self.head is None:
        self.head=node(val)
        self.last=self.head
    else:
        while self.head.befor:
            self.head=self.head.befor
        self.head.befor=node(val)
        self.head.befor.after=self.head
        self.head=self.head.befor
        self.head=self.last
def add_column(self,index,val):
    if index==1:
        if self.head is None:
            self.head=node(val)
            self.last=self.head
        else:
            while self.head.next:
                self.head=self.head.next
            self.head.next=node(val)
            self.head.next.seco=self.head
            self.head=self.head.next
            self.head=self.last
    else:

```

```

        while index>1:
            self.head=self.head.befor
            index-=1
        while self.head.next:
            self.head=self.head.next
        self.head.next=node(val)
        self.head.next.seco=self.head
        self.head=self.head.next
        self.head=self.last
def show(self):
    while self.head is not None:
        nod=self.head.befor
        while self.head is not None:
            print(self.head.data,end=" ")
            self.head=self.head.next
        self.head=nod
        print(" ")
    self.head=self.last

```

```

l=linked()
l.add_row(10)
l.add_row(11)
l.add_row(12)
l.add_column(1,10)
l.add_column(1,11)
l.add_column(1,12)
l.add_column(2,13)

```

```
l.add_column(2,13)
l.add_column(2,14)
l.add_column(2,15)
l.add_column(3,16)
l.add_column(3,17)
l.add_column(3,18)
l.add_column(1,13)
l.add_column(1,14)
```

```
l.show()
```

```
=====
```

```
*****
المشروع هذا اذا كان هنالك نقص بالدرجات وهذا المشروع عبارة عن لعبة الشطرنج
*****
```

```
ملف الهيدر الذي يحتوي على دوال تقوم بتحسين وترتيب الواجهة
*****
```

```
#pragma once
#include<iostream>
#include<thread>
#include<chrono>
#include<Windows.h>
using namespace std;
```

```
#pragma once
#include<iostream>
#include<thread>
#include<chrono>
#include<Windows.h>
using namespace std;

int cou[8][8];
char name_class, con = ' ';
int n = 1, N, z = 0;
int tr, fo;
char First_player[16], Second_player[16];
static int Count_First_player = 0;
static int Count_Second_player = 0;

void Print_the_first_spoils() {
    cout << " ";
    for (int i = 0; i < Count_First_player; i++)
        cout << First_player[i] << " ";
    cout << endl;
}

void Print_the_second_spoils() {
    cout << " ";
    for (int i = 0; i < Count_Second_player; i++)
        cout << Second_player[i] << " ";
    cout << endl;
}
```

```

}

void line(void) {
    cout << " ";
    for (int i = 0; i < 32; i++)
        cout << "-";
    cout << endl;
}

void linecount(void) {
    cout << " ";
    for (int i = 0; i < 8; i++) {
        cout << " " << i;
    }
    cout << "\n";
}

void Stop_code()
{
    Beep(1000, 500);
    this_thread::sleep_for(chrono::seconds(3));
}

void Screen_filter(void) { system("cls"); }

```

```

*****
الملف التنفيذي
*****

```

```
#include<iostream>
#include"Coordination_function_and_variables.h"
using namespace std;
class class {
protected:
    char ch;
public:
    chass() {}
    chass(char c) :ch(c) {}
    void setch(char ch) { this->ch = ch; }
    char getch() { return this->ch; }

    int condition() {
        if (this->ch >= 'A'&&this->ch <= 'F')
            return 1;
        else if (this->ch >= 'a'&&this->ch <= 'f')
            return 0;
    }
    void set_castel(class *arr[][8], int x, int y) {
        if (name_class >= 'A'&&name_class <= 'F') { tr = 1; }
        else if (name_class >= 'a'&&name_class <= 'f') { tr = 0; }
        for (int i = x; i <= 7; i++) {
            z = 0;
            for (int j = y; j <= 7; j++) {
                if ((i == x&&j >= y + 1) || (i >= x + 1 && j == y))
                    if (arr[i][j]->getch() == ' ')
                        cout<<endl;
            }
        }
    }
};
```



```

        else if (arr[i][j]->getch() != ' ') {
            if (arr[i][j]->condition() == tr) {
                if (i >= x + 1 && j == y)
                    z = 1;
                break;
            }
            else {
                cou[i][j] = n++;
                if (i >= x + 1 && j == y)
                    z = 1;
                break;
            }
        }
        else
            continue;
    }
    if (z == 1)
        break;
}
for (int i = y - 1; i >= 0; i--)
    if (arr[x][i]->getch() == ' ')
        cou[x][i] = n++;
    else if (arr[x][i]->getch() != ' ') {
        if (arr[x][i]->condition() == tr)
            break;
        else {
            cou[x][i] = n++;

```

```

        break;
    }
}
for (int i = x - 1; i >= 0; i--)
    if (arr[i][y]->getch() == ' ')
        cou[i][y] = n++;
    else if (arr[i][y]->getch() != ' ') {
        if (arr[i][y]->condition() == tr)
            break;
        else {
            cou[i][y] = n++;
            break;
        }
    }
}

void set_officer(chass *arr[][8], int x, int y) {
    if (name_class >= 'A' && name_class <= 'F') { tr = 1; fo = 0; }
    else if (name_class >= 'a' && name_class <= 'f') { tr = 0; fo = 1; }
    z = 0;
    for (int i = x + 1; i <= 7; i++)
    {
        for (int j = y + 1; j <= 7; j++)
            if (i + (y - x) == j && arr[i][j]->getch() == ' ')
                cou[i][j] = n++;
            else if (i + (y - x) == j && arr[i][j]->getch() != ' ' && arr[i][j]->condition() == tr) {
                z = 1;
                break;
            }
    }
}

```

```

    }
    else if (i + (y - x) == j && arr[i][j]->getch() != ' ' && arr[i][j]->condition() == fo) {
        cou[i][j] = n++;
        z = 1;
        break;
    }
    if (z == 1)
        break;
}
z = 0;
for (int i = x - 1; i >= 0; i--)
{
    for (int j = y - 1; j >= 0; j--)
        if (i + (y - x) == j && arr[i][j]->getch() == ' ')
            cou[i][j] = n++;
    else if (i + (y - x) == j && arr[i][j]->getch() != ' ' && arr[i][j]->condition() == tr) {
        z = 1;
        break;
    }
    else if (i + (y - x) == j && arr[i][j]->getch() != ' ' && arr[i][j]->condition() == fo) {
        cou[i][j] = n++;
        z = 1;
        break;
    }
    if (z == 1)
        break;
}
}

```

```

z = 0;
for (int i = x + 1; i <= 7; i++)
{
    for (int j = y - 1; j >= 0; j--)
        if (i + j == y + x && arr[i][j]->getch() == ' ')
            cou[i][j] = n++;
        else if (i + j == y + x && arr[i][j]->getch() != ' ' && arr[i][j]->condition() == tr) {
            z = 1;
            break;
        }
        else if (i + j == y + x && arr[i][j]->getch() != ' ' && arr[i][j]->condition() == fo) {
            cou[i][j] = n++;
            z = 1;
            break;
        }
        if (z == 1)
            break;
}
z = 0;
for (int i = x - 1; i >= 0; i--)
{
    for (int j = y + 1; j <= 7; j++)
        if (i + j == y + x && arr[i][j]->getch() == ' ')
            cou[i][j] = n++;
        else if (i + j == y + x && arr[i][j]->getch() != ' ' && arr[i][j]->condition() == tr) {
            z = 1;
            break;
        }
}

```

```

    }
    else if (i + j == y + x && arr[i][j]->getch() != ' ' && arr[i][j]->condition() == fo) {
        cou[i][j] = n++;
        z = 1;
        break;
    }
    if (z == 1)
        break;
}
}
virtual void setchass(chass *arr[][8], int x, int y) {}
};

```

```

void print(chass *arr[][8]);
void view_available_sites(int x, int y, chass *arr[][8], char name_class);

```

```

class A :public chass {
public:
    A(char c) :chass(c) {}
    void setchass(chass *arr[][8], int x, int y) {
        print(arr);
        name_class = arr[x][y]->getch();
        arr[x][y]->set_castel(arr, x, y);
        print(arr);
        view_available_sites(x, y, arr, name_class);
        print(arr);
        n = 1;
    }
};

```

```

    }
};

class B :public chass {
public:
    B(char c) :chass(c) {}
    void setchass(chass *arr[][8], int x, int y)
    {
        name_class = arr[x][y]->getch();
        if (name_class >= 'A' && name_class <= 'F') { tr = 0; }
        else if (name_class >= 'a' && name_class <= 'f') { tr = 1; }
        for (int i = x - 2; i <= x + 2; i++)
            for (int j = y - 2; j <= y + 2; j++)
                if (((i == x - 2 || i == x + 2) && (j == y - 1 || j == y + 1)) || ((i == x - 1 || i == x +
1) && (j == y - 2 || j == y + 2))) && ((i >= 0 && i <= 7) && (j >= 0 && j <= 7)))
                    if (arr[i][j]->getch() == ' ')
                        cou[i][j] = n++;
                    else if (arr[i][j]->getch() != ' ' && arr[i][j]->condition() == tr)
                        cou[i][j] = n++;

        print(arr);
        view_available_sites(x, y, arr, name_class);
        print(arr);
        n = 1;
    }
};

class C :public chass {

```

```
public:
    C(char c) :chass(c) {}
    void setchass(chass *arr[][8], int x, int y)
    {
        name_class = arr[x][y]->getch();
        arr[x][y]->set_officer(arr, x, y);
        print(arr);
        view_available_sites(x, y, arr, name_class);
        print(arr);
        n = 1;
    }
};
```

```
class D :public chass {
public:
    D(char c) :chass(c) {}
    void setchass(chass *arr[][8], int x, int y)
    {
        name_class = arr[x][y]->getch();
        arr[x][y]->set_castel(arr, x, y);
        arr[x][y]->set_officer(arr, x, y);
        print(arr);
        view_available_sites(x, y, arr, name_class);
        print(arr);
        n = 1;
    }
};
```

```

class E :public chass {
public:
    E(char c) :chass(c) {}
    void setchass(chass *arr[][8], int x, int y)
    {
        name_class = arr[x][y]->getch();
        if (name_class >= 'A' && name_class <= 'F') { tr = 1; fo = 0; }
        else if (name_class >= 'a' && name_class <= 'f') { tr = 0; fo = 1; }
        for (int i = x - 1; i <= x + 1; i++)
            for (int j = y - 1; j <= y + 1; j++)
            {
                if (i >= 0 && j >= 0 && i <= 7 && j <= 7)
                    if (arr[i][j]->getch() == ' ')
                        cou[i][j] = n++;
                    else if (arr[i][j]->getch() != ' ' && arr[i][j]->getch() != name_class)
                        if (arr[i][j]->condition() == tr)
                            continue;
                        else if (arr[i][j]->condition() == fo) { cou[i][j] = n++; }
            }
        print(arr);
        view_available_sites(x, y, arr, name_class);
        print(arr);
        n = 1;
    }
};

class F :public chass {

```



```

class F :public chass {
public:
    F(char c) :chass(c) {}
    void setchass(chass *arr[][8], int x, int y)
    {
        name_class = arr[x][y]->getch();
        int row;
        if (name_class == 'F') { tr = 1; row = x + 1; }
        else if (name_class == 'f') { tr = 0; row = x - 1; }
        if (arr[row][y]->getch() == ' ')
            cou[row][y] = n++;
        if (arr[row][y + 1]->getch() != ' ' && y == 0 && !arr[row][y + 1]->condition() == tr)
            cou[row][y + 1] = n++;
        if (arr[row][y - 1]->getch() != ' ' && y == 7 && !arr[row][y - 1]->condition() == tr)
            cou[row][y - 1] = n++;
        if (y - 1 >= 0 && y + 1 <= 7 && !arr[row][y - 1]->condition() == tr && arr[row][y - 1]->getch() != ' ')
            cou[row][y - 1] = n++;
        if (y - 1 >= 0 && y + 1 <= 7 && !arr[row][y + 1]->condition() == tr && arr[row][y + 1]->getch() != ' ')
            cou[row][y + 1] = n++;
        print(arr);
        view_available_sites(x, y, arr, name_class);
        print(arr);
        n = 1;
    }
};

void print(chass *arr[][8])

```

```

void print(chass *arr[][8])
{
    Screen_filter();
    Print_the_first_spoils();
    line();
    linecount();
    line();
    for (int i = 0; i<8; i++) {
        cout << " " << i;
        for (int j = 0; j<8; j++)
            if (j == 0)
                cout << "|" << " " << arr[i][j]->getch();
            else
                cout << " " << arr[i][j]->getch();
        cout << "|\\n";
    }
    line();
    Print_the_second_spoils();
    line();
    linecount();
    line();
    for (int i = 0; i < 8; i++) {
        cout << " " << i;
        for (int j = 0; j < 8; j++)
            if (j == 0)
                if (cou[i][j] <= 9)
                    cout << "|" << "0" << cou[i][j];
    }
}

```

```

        else
            cout << "|" << cou[i][j];
    else
        if (cou[i][j] <= 9)
            cout << " " << "0" << cou[i][j];
        else
            cout << " " << cou[i][j];
    cout << "|\n";
}
line();
}

```

```

void view_available_sites(int x, int y, chass *arr[][8], char name_class) {
    cout << "    enter location: ";
    cin >> N;
    for (size_t i = 0; i < 8; i++)
    {
        for (size_t Y = 0; Y < 8; Y++)
        {
            if (cou[i][Y] == N&&N != 0) {
                N = 99;
                con = name_class;
                if (name_class >= 'A'&&name_class <= 'F'&&arr[i][Y]->getch() != ' ')
                {
                    First_player[Count_First_player++] = arr[i][Y]->getch(); system("color 5F");
                    Stop_code();
                }
            }
        }
    }
}

```

```
,
else if (name_class >= 'a' && name_class <= 'f' && arr[i][Y] -> getch() != ' ')
{
    Second_player[Count_Second_player++] = arr[i][Y] -> getch(); system("color 5F");
    Stop_code();
}
delete arr[x][y];
delete arr[i][Y];
arr[x][y] = new chass(' ');
if (name_class == 'a' || name_class == 'A')
    arr[i][Y] = new A(name_class);
else if (name_class == 'b' || name_class == 'B')
    arr[i][Y] = new B(name_class);
else if (name_class == 'c' || name_class == 'C')
    arr[i][Y] = new C(name_class);
else if (name_class == 'd' || name_class == 'D')
    arr[i][Y] = new D(name_class);
else if (name_class == 'e' || name_class == 'E')
    arr[i][Y] = new E(name_class);
else if (name_class == 'f' || name_class == 'F')
    arr[i][Y] = new F(name_class);
cou[i][Y] = 0
```

```
#include<iostream> //برنامچ لرترب مكدس
using namespace std;
int size=10,*a=new int[size],top=-1;
static int x=0;
void push(int a[],int k){
if(top==size-1){
cout<<"full stak\n";
return;}
else
a[++top]=k;
}

int pop(){
if(top<0){
cout<<"empty stak\n";
return 0;}
else
return a[top--];
}

void print(){
for(int i=top;i>=0;i--)
cout<<a[i]<<endl;
}

void maxes(){
```

```

void maxes(){
int min;
for(int i=top;i>=0;i--){
min=0;
for(int j=i-1;j>=0;j--){
if(a[i]<a[j]){
min=a[j];
a[j]=a[i];
a[i]=min;
}
}
}
}

int main()
{
int k;
for(int i=0;i<size;i++){
cin>>k;
push(a,k);
}
cout<<"the after\n";
print();
maxes();
cout<<"the secod to maxes\n";
print();
}

```

```
a[i]=min,  
}  
}  
}  
}  
  
int main()  
{  
int k;  
for(int i=0;i<size;i++){  
cin>>k;  
push(a,k);  
}  
cout<<"the after\n";  
print();  
maxes();  
cout<<"the secod to maxes\n";  
print();  
  
for(int i=0;i<size;i++)  
cout<<pop()<<endl;  
return 0;  
}
```

برنامج اضافة طلاب بحيث يكون اخر طالب اسمة مكون من اول حرف لكل طالب وعمره مجموع اعمارهم وطولة مجموع اطوالهم

```
#include<iostream>
#include<string.h>
using namespace std;
const int size=4;
struct student{
char name[15];
int age;
float lngth;
};
void add_student(student s[])
{
for(int i=0;i<size-1;i++){
cout<<"enter name student "<<i+1<<endl;
cin>>s[i].name;
cout<<"enter age student "<<i+1<<endl;
cin>>s[i].age;
cout<<"enter lngth student "<<i+1<<endl;
cin>>s[i].lngth;
}
}
void print(student *s){
s[size-1].age=0;
```



```
s[size-1].length=0;
for(int i=0;i< size-1;i++){
    strncat(s[size-1].name,s[i].name,1);
    s[size-1].age+=s[i].age;
    s[size-1].length+=(s[i].length/3);
    switch(i){
    case 0:
        cout<<s[i].name<<" ";
        break;
    case 1:
        cout<<s[i].age<<" ";
        break;
    case 2:
        cout<<s[i].length<<endl;
        break;}}
}
int main()
{
    student s1[size];
    add_student(s1);
    print(s1);
    return 0;
}
```

طباعة القطرين معى صفين وعمودين من المنتصف

```
#include<iostream>
using namespace std;
char a[20][20];
```

```
void setaa(){
for(int i=0;i<20;i++)
for(int j=0;j<20;j++)
a[i][j]='*';
}
```

```
void print(){
for(int i=0;i<20;i++){
for(int j=0;j<20;j++){
if(i==j||j+i==19)
cout<<a[i][j];
else if(j==(20/2-1)||j==(20/2+1))
cout<<a[i][j];
else if(i==(20/2-1)||i==(20/2+1))
cout<<a[i][j];
}
```

```
cout<<a[i][j];  
else if(j==(20/2-1)||j==(20/2+1))  
cout<<a[i][j];  
else if(i==(20/2-1)||i==(20/2+1))  
cout<<a[i][j];  
else cout<<" ";  
}  
cout<<endl;  
}  
}
```

```
int main()  
{  
setaa();  
print();  
return 0;  
}
```

الصفحة ١،٢ برنامج لادخال درجات طلاب وطباعة مجموعهم ومعدلهم وتقديرهم
الصفحة ٣،٤ وجزى من الصفحة ٥ برنامج لتطبيق القائمة المترابطة
الصفحة ٥،٦ برنامج لاختبار العدد اولي او زوجي او فردي
من بداية (جزى من الصفحة ٦ الى نصف الصفحة عشرة) برنامج يحاكي الالة الحاسبة
من نصف الصفحة ١٠ الى نصف الصفحة ١٣ برنامج يعمل على انشاء مصفوفة ثنائية البعد ديناميكية