CS221 - Lab 8 : Dynamic Arrays

Objective(s)

1. Solving problems with pointers and dynamic variables.
2. Using multidimensional dynamic arrays.

Tool(s)/Software

DevC++.

Tasks/Assignments(s)

Task- 1: [Solve it yourself without using the C++]

Give the value of the left-hand side variable in each assignment statement. Assume the lines are executed sequentially. Assume the address of the blocks array is 4434. Show what's happening in the memory.

```
int main()
{
   char blocks[3] = {'A','B','C'};
   char *ptr = blocks;
   char temp;
   temp = blocks[0];
   temp = *(blocks + 2);
   temp = *(ptr + 1);
   temp = *ptr;

   ptr = blocks + 1;
   temp = *ptr;
   temp = *(ptr + 1);

   ptr = blocks;
   temp = *++ptr;
   temp = ++*ptr;
   temp = *ptr++;
   temp = *ptr;
   return 0;
}
```

Solution:

```
   char blocks[3] = {'A','B','C'};   //try int blocks[3]={10,20,30};
   char *ptr =blocks;
   char temp;
   temp = blocks[0];    //A, 10
   cout<<temp<<endl;
   temp = *(blocks + 2);  //C, 30
   cout<<temp<<endl;
   temp = *(ptr + 1);  //B, 20
   cout<<temp<<endl;
   temp = *ptr;        //A,10
   cout<<temp<<endl;

   ptr = blocks + 1;  //&b[1]  it printed all array elements starting from
b[1] BC but it will print the address for non-char array
   cout<<ptr<<endl;
   temp = *ptr;  //B, 20
   cout<<temp<<endl;
   temp = *(ptr + 1);  //C, 30
   cout<<temp<<endl;

   ptr = blocks;  //&b[0] //ABC
   cout<<ptr<<endl;
   temp = *++ptr; //B, 20
   cout<<temp<<endl;
   temp = ++*ptr; // b+1=C , 21
   cout<<temp<<endl;
   temp = *ptr++; //C, 21  cause the value of b[1] has changed to C (take the
value & save it in temp THEN move the pointer
   cout<<temp<<endl;
   temp = *ptr;  //C, 30
   cout<<temp<<endl;
   return 0;
```

Task #2 Calculate the Body Mass Index (BMI)

The body mass index is calculated by dividing the weight (in kilograms) over the height squared (in meters). A healthy BMI ranges between 19 and 25. Ask the user to enter their weight and height and calculate the user's BMI using the formula, $BMI = \frac{Weight\ (km)}{Height^2\ (cm)} \times 10000$ and tell them their result whether they are healthy or not.

Constraints:

1- Use dynamic variables to solve this question
2- Delete the dynamic variables when you no longer need to use them.
3- Use the following function to calculate the BMI and return the result:
double BMI (double *weight, double *height);

Sample Run:

```
Enter your weight and height 45 1.53

healthy

Enter a positive number to continue1

Enter your weight and height 65 1.75

healthy

Enter a positive number to continue1

Enter your weight and height 45 1.80

unhealthy

Enter a positive number to continue-1
```

Solution:

```
//Lab8-Task2 BMI -no loop

#include<cmath>
#include<iostream>
using namespace std;

double BMI(double *w, double *h ){ return (*w/ pow(*h,2))*10000; }

int main() {
    double *weight,*height, result;

    weight = new double;
    height = new double ;

    cout << "Plz. Enter your weight (kg): ";
    cin >> *weight ;
    cout << "Plz. Enter your height (cm): ";
    cin >> *height ;

    result=BMI(weight,height);
    cout<< "\nResult: "<< result;

    if (result>=19 && result<=25)
       cout<<"\tHealthy";
    else
       cout<<"\tUnhealthy";   //there are more cases
(underweight/overweight/obese)

 // Deallocate pointers
    delete weight;   weight=NULL;
    delete height;   height=NULL;

    return 0;
} //end of main
```

Task- 3

Sara loves her coffee, she buys several cups every day. She also doesn't like to buy from the same coffee shop, thus each day her coffee might have a different price. Write a C++ program to calculate the total price she paid this week.

1. Create 2 dynamic arrays in the main function, one to store the costs and the other to store the number of cups bought per day.
2. Use a function to fill the arrays with the number of cups she bought and the cost of each cup.

   void fillArray(double* cost, int* cups, int arraySize)

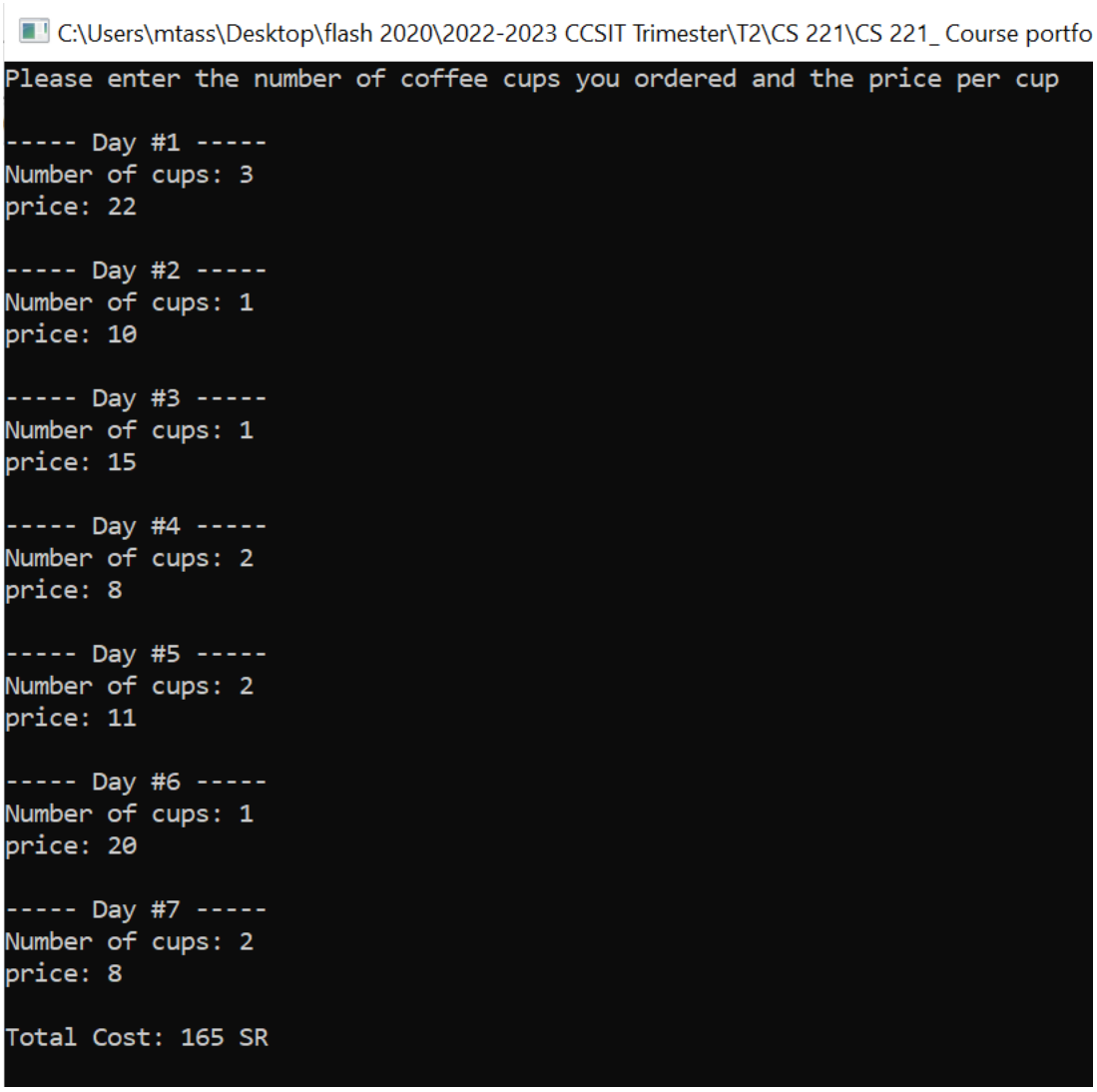
3. Use a function to calculate the total price she paid this week.

   double totalCost(double* cost,int* cups,int arraySize)

4. Deallocate the dynamically allocated arrays.

Hint: the arrays will each have 7 cells, since there are 7 days in a week.

Sample Run:



Solution:

```
//Lab8-task3 Daily Coffee Calculator

#include<iostream>
using namespace std;

double totalCost(double* cost,int* cups,int arraySize)
{
    double total=0;
    for(int i=0;i<arraySize;i++)
           total+=cups[i]*cost[i];

    return total;
}//end totalCost

void fillArray(double* cost,int* cups, int arraysize)
{
    cout<<"Please enter the number of coffee cups you ordered and the price
per cup\n\n";

    for(int i=0;i<arraysize;i++){
        cout<<"----- Day #"<<i+1<<" -----\n"<<"Number of cups: ";
        cin>>cups[i];
        cout<<"price: ";
        cin>>cost[i];
        cout<<endl;
    }//end for
}//end fillArray

int main()
{
    int week=7;  //number of days
    double* cost=new double[week];
    int* cups=new int[week];

    fillArray(cost,cups,week);

    cout<<"Total Cost: "<<totalCost(cost,cups,week)<<" SR"<<endl;

    delete []cost;  cost=NULL;
    delete []cups;  cups=NULL;

    return 0;
}//end main
```

Task-4 Airplane Reservation System

The Saudi airlines need your help in their reservation system. Write a C++ program that does the following:

1. Allow the user to enter the size of the airplane in terms of rows and columns
2. Create a dynamic 2D array of Boolean type with the specified rows and columns.

3. Display a menu to the user with the following tasks:
   a. Reserve a seat

      void reserve(bool **arr,int rows,int cols)

      This function allows the user to enter the row and column of the seat they would like to reserve. If the seat is available, it is then reserved for the user. Otherwise, a message is displayed informing the user with the non-availability of the requested seat.

   b. Display number of available seats

      void available(bool **arr,int rows,int cols)

      This function displays the number of available seats which have not been reserved yet.

   c. Display all seats in the plane

      void display(bool **arr,int rows,int cols)

      This function visualizes to the user all the seats in the plane. Reserved seats are displayed "1" and unreserved seats are displayed "0".

4. The program should continue until the user enters 'n'.
5. Finally, delete the dynamic array when it's no longer used.

Sample Run:

```
How many rows are in the plane 2
How many seats per row 2
----menu----
1-reserve a seat
2-display the available seats
3-display all the seats in the plane
2
available seats: 4

Would you like to quit?n
----menu----
1-reserve a seat
2-display the available seats
3-display all the seats in the plane
3
0 0
0 0

Would you like to quit?n
----menu----
1-reserve a seat
2-display the available seats
3-display all the seats in the plane
1
which seat would you like to reserve [row][col]1 1
The seat has been reserved!
Would you like to quit?n
```

----menu----
1-reserve a seat
2-display the available seats
3-display all the seats in the plane
1
which seat would you like to reserve [row][col]0 1
The seat has been reserved!
Would you like to quit?n
----menu----
1-reserve a seat
2-display the available seats
3-display all the seats in the plane
3
0 1
0 1

Would you like to quit?n
----menu----
1-reserve a seat
2-display the available seats
3-display all the seats in the plane
1
which seat would you like to reserve [row][col]0 1
Sorry, the seat is already reserved
Would you like to quit?n
----menu----
1-reserve a seat
2-display the available seats
3-display all the seats in the plane
1
which seat would you like to reserve [row][col]1 0
The seat has been reserved!
Would you like to quit?n
----menu----
1-reserve a seat
2-display the available seats
3-display all the seats in the plane
1
which seat would you like to reserve [row][col]0 0
The seat has been reserved!
Would you like to quit?n
----menu----
1-reserve a seat
2-display the available seats
3-display all the seats in the plane
1
The plane is full
Would you like to quit?n
----menu----
1-reserve a seat
2-display the available seats
3-display all the seats in the plane
3
1 1
1 1

Would you like to quit?y
Program ended with exit code: 0

Solution:

```
/*
       lab8 task4 Airplane Reservation System
       Author: Mona Altassan
*/
#include <iostream>
using namespace std;

int available(bool **arr,int rows,int cols)
{
    int count=0;

    for(int i=0;i<rows;i++)
        for(int j=0;j<cols;j++)
            if(!arr[i][j])
               count++;

    return count;
}//end of available

void reserve(bool **arr,int rows,int cols)
{
    int row,col;

    //check if the plan is full
    if(available(arr, rows, cols)==0)
    {  cout<<"\nSorry the plane is full!!";
       return;
    }//end if

    cout<<"\nWhich seat would you like to reserve [row][col]";
    cin>>row>>col;

    if(row>rows || col>cols || row<0 || col<0)
    {
        cout<<"\nERROR: Invalid row/column value!!!";
        return;
    }//end if

    //check if the requested seat is already reserved i.e. true/1
    if(!arr[row][col]) //arr[r][c]==false/0
    {
          arr[row][col]=true;
       cout<<"\nThe seat has been reserved!";
    }
      else
       cout<<"\nSorry, the seat is already reserved";

}//end of reserve
```

```
void display(bool **arr,int rows,int cols)
{
    for(int i=0;i<rows;i++)
       {
        for(int j=0;j<cols;j++)
            if(!arr[i][j])
                cout<<"0"<<" ";
            else
                cout<<"1"<<" ";

        cout<<endl;
    }//end outer for
}//end of display

int main()
{
    int rows,cols,choice;
    char quit;

    //specify size of arrays
    cout<<"\n**** Airplane Reservation System ****\n\n";
    cout<<"How many rows in the plane? ";
    cin>>rows;
    cout<<"How many seats per row? ";
    cin>>cols;

    //initialize arrays (default=0 i.e. not reserved)
    bool **arr=new bool*[rows];
    for(int i=0;i<rows;i++)
        arr[i]=new bool[cols];

    //Airline System Tasks
    do{
        cout<<"\n------ Menu ------\n1-Reserve a seat \n2-Display available
seats\n3-Display all seats\n>> ";
        cin>>choice;

        if(choice==1)
           reserve(arr,rows,cols);
        else if (choice ==2)
           cout<<"\nAvailable seats: "<<available(arr,rows,cols);
        else if (choice==3)
           display(arr,rows,cols);
        else
           cout<<"\nERROR: Invalid choice!!";

        cout<<"\n\nWould you like to quit (y/n)? ";
        cin>>quit;
     }while(tolower(quit)=='n');

    //deallocate pointer
    delete [] arr;   arr=NULL;
    return 0;
}//end of main
```