**LAB 02**

**QUESTION 1:**

**1. Write a function that dynamically allocates an array of integers. The function should accept an**

**integer argument indicating the number of elements to allocate. The function should return a**

**pointer to the array.**

**Write a function named coin Toss that simulates the tossing of a coin. When you call the function, it should generate a random number in the range of 1 through 2. If the random number is 1, the function should display “heads.” If the random number is 2, the function should display “tails.” Demonstrate the function in a program that asks the user how many times the coin should be tossed and then simulates the tossing of the coin that number of times.**

**PROGRAM**:

#include<iostream>

using namespace std;

int \*dma(int *n*){

    int \*arr=new int[*n*];

    if(arr!=NULL){

        cout<<"array allocated sucessfully"<<endl;

    }

    return arr;

}

int main(){

    int n;

    cout<<"Enter how many element you in an array :";

    cin>>n;

int \*array=dma(n);

    for(int i=0;i<n;i++){

        cout<<"Enter element "<<i+1<<" :";

        cin>>array[i];

    }

    for(int i=0;i<n;i++){

        cout<<array[i]<<" ";

    }

    delete[] array;

}

**RESULT**:

A screenshot of a computer

Description automatically generated

**QUESTION#2**

**In statistics, the mode of a set of values is the value that occurs most often or with the greatest frequency. Write a function that accepts as arguments the following: A) An array of integers B )An integer that indicates the number of elements in the array The function should determine the mode of the array. That is, it should determine which value in the array occurs most often. The mode is the value the function should return. If the array has no mode (none of the values occur more than once), the function should return -1. (Assume the array will always contain nonnegative values.) Demonstrate your pointer prowess by using pointer notation instead of array notation in this function.**

#include <iostream>

using namespace std;

int mode(int n,int \*arr){

    int count[n]={0};

    int index=0;

for(int i=0;i<n;i++){

        count[arr[i]]++;

    }

    int maximum=count[0];

    for(int i=0;i<n;i++){

    if(maximum < count[i]){

        maximum=count[i];

        index=i;

    }

}

if(maximum>2){

    return index;

}

else{

    return -1;

}

}

int main(){

    int n;

    cout<<"Enter number of element you want to enter : ";

    cin>>n;

    int arr[n];

    for(int i=0;i<n;i++){

        cout<<"Enter element "<<i<<" :";

        cin>>arr[i];

    }

    int \*ptr=arr;

    int repeat=mode(n,arr);

    if(repeat==-1){

        cout<<"there is not mode"<<endl;

    }

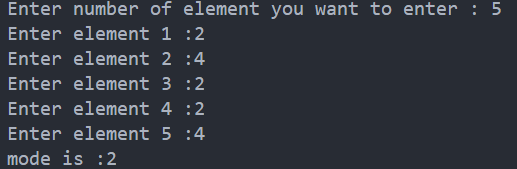
    else{

        cout<<"mode is :"<<repeat<<endl;

    }

}

**RESULT**:



**QUESTION#3**

**Write a program that count the number of vowels and consonants in a string using a pointer.**

#include<iostream>

using namespace std;

void  volcons(char \*ptr,string word){

     int vowel=0; int consonant=0;

    for(int i=0;i<word.length();i++){

        if((\*ptr+i > 'a' && \*ptr + i <'z')|| (\*ptr + i >'A' && \*ptr + i <'Z') )

        if(\*(ptr + i)=='a'||\*(ptr + i)=='e' ||  \*(ptr + i)=='i' ||  \*(ptr + i)=='o' || \*(ptr + i)=='u' ||\*(ptr + i)=='A'|| \*(ptr + i)=='E' || \*(ptr + i)=='I' ||  \*(ptr + i)=='O' || \*(ptr + i)=='U' ){

            vowel++;

        }

        else{

            consonant++;

        }

    }

    cout<<"vowel is :"<<vowel<<endl;

    cout<<"consonent is :"<<consonant<<endl;

}

int main(){

    string word;

    cout<<"enter string :";

    cin>>word;

    char \*ptr=&word[0];

    volcons(ptr,word);

}

**RESULT:**

**A black background with white text

Description automatically generated**

**QUESTION#4**

**Write a program that implements a basic calculator using pointer arithmetic to perform addition,subtraction, multiplication, and division. The program should take two integer inputs from the user, perform the selected operation, and output the result. Use pointers to pass values to functions and handle the result.**

#include<iostream>

using namespace std;

int calculator(int \*a,int \*b){

    char op;

    do{

    cout<<"enter operation :"<<endl;

    cin>>op;

          switch(op){

            case  '+' :

                return \*a+\*b;

                break;

            case '-' :

                return \*a-\*b;

                break;

            case '/' :

                return \*a/ (\*b);

                break;

            case '\*' :

                return (\*a)\*(\*b);

                break;

            default:

                cout<<"invalid operator, please enter  again :";

                break;

          }

    }   while(op != '\*'&& op !='+' && op !='-' &&  op !='/');

}

int main(){

     int a; int b;

    cout<<"enter first value :";

    cin>>a;

    cout<<"enter second value :";

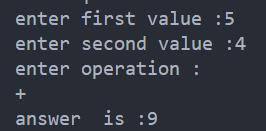
    cin>>b;

    int answer=calculator(&a,&b);

    cout<<"answer  is :"<<answer<<endl;

}

**RESULT**:



**QUESTION#5**

**Write a program that uses a structure named MovieData to store the following information about**

**a movie:**

**Title**

**Director**

**Year Released**

**Running Time (in minutes)**

**The program should create two MovieData variables, store values in their members, and pass each one, in turn, to a function that displays the information about the movie in a clearly formatted manner.**

#include<iostream>

using namespace std;

    struct MovieData {

        string title;

        string director;

        int runtime;

        int yearReleased;

    };

void display(struct MovieData m){

        cout<<"movie title :"<<m.title<<endl;

        cout<<"movie director :"<<m.director<<endl;

        cout<<"movie runtime :"<<m.runtime<<endl;

        cout<<"movie yearReleased :"<<m.yearReleased<<endl<<endl;

}

int main(){

 struct MovieData m1;

 struct  MovieData m2;

    cout<<"Enter for Movie 1"<<endl;

        cout<<"Enter Movie title :";

        cin>>m1.title;

        cout<<"Enter Movie director :";

        cin>>m1.director;

        cout<<"Enter runtime  :";

        cin>>m1.runtime;

        cout<<"Enter Movie year released :";

        cin>>m1.yearReleased;

        cout<<"Enter for Movie 2"<<endl;

        cout<<"Enter Movie title :";

        cin>>m2.title;

        cout<<"Enter Movie director :";

        cin>>m2.director;

        cout<<"Enter runtime  :";

        cin>>m2.runtime;

        cout<<"Enter Movie year released :";

        cin>>m2.yearReleased;

    cout<<"Movie 1 is :"<<endl;

display(m1);

    cout<<"Moive 2 is :"<<endl;

    display(m2);

**}**

**RESULT**:

**A screen shot of a computer

Description automatically generated**

**QUESTION#6:**

**Write a program that uses a structure to store the following data about a customer account:**

**Name**

**Address**

**City, State, and ZIP**

**Telephone Number**

**Account Balance**

**Date of Last Payment**

**The program should use an array of at least 10 structures. It should let the user enter data into the array, change the contents of any element, and display all the data stored in the array. The program should have a menu-driven user interface.**

**Input Validation: When the data for a new account is entered, be sure the user enters data for all the fields. No negative account balances should be entered.**

#include <iostream>

#include <string>

using namespace std;

 int SIZE = 10;

struct Customer {

    string name;

    string address;

    string city\_state\_zip;

    string number;

    float balance;

    string payment;

};

void add\_customer(Customer *c*[], int &*count*) {

    if (*count* >= SIZE) {

        cout <<endl<<"Customer list is full"<<endl;

        return;

    }

    cout << "Enter Name: ";

    cin >> *c*[*count*].name;

    cout << "Enter Address: ";

    cin >> *c*[*count*].address;

    cout << "Enter City, State, ZIP: ";

    cin >> *c*[*count*].city\_state\_zip;

    cout << "Enter Phone Number: ";

    cin >> *c*[*count*].number;

    do {

        cout << "Enter Account Balance: "<<endl;

        cin >> *c*[*count*].balance;

        if (*c*[*count*].balance < 0)

            cout << "Invalid number. Try again"<<endl;

    } while (*c*[*count*].balance < 0);

    cout << "Enter Last Payment Date: ";

    cin >> *c*[*count*].payment;

*count*++;

    cout << "Customer added successfully!"<<endl;

}

void edit\_customer(Customer *c*[], int *count*) {

    if (*count* == 0) {

        cout <<endl<<"No customers to edit."<<endl;

        return;

    }

    int index;

    cout << "Enter customer index (0 to " << *count* - 1 << "): ";

    cin >> index;

    if (index < 0 || index >= *count*) {

        cout << "Invalid index."<<endl;

        return;

    }

    cout << "Enter New Name: ";

    cin >> *c*[index].name;

    cout << "Enter New Address: ";

    cin >> *c*[index].address;

    cout << "Enter New City, State, ZIP: ";

    cin >> *c*[index].city\_state\_zip;

    cout << "Enter New Phone Number: ";

    cin >> *c*[index].number;

    do {

        cout << "Enter New Account Balance: ";

        cin >> *c*[index].balance;

        if (*c*[index].balance < 0)

            cout << "Account balance cannot be negative. Try again.\n";

    } while (*c*[index].balance < 0);

    cout << "Enter New Last Payment Date: ";

    cin >> *c*[index].payment;

    cout << "Customer updated successfully!"<<endl;

}

void display\_customer(const Customer *c*[], int *count*) {

    if (*count* == 0) {

        cout << "\nNo customers to display.\n";

        return;

    }

    for (int i = 0; i < *count*; i++) {

        cout << "\nCustomer " << i << endl;

        cout << "Name: " << *c*[i].name <<endl;

        cout << "Address: " << *c*[i].address << endl;

        cout << "City, State, ZIP: " << *c*[i].city\_state\_zip << endl;

        cout << "Phone Number: " << *c*[i].number <<endl;

        cout << "Account Balance:" << *c*[i].balance << endl;

        cout << "Last Payment Date: " << *c*[i].payment << endl;

    }

}

int main() {

    Customer c[SIZE];

    int count = 0;

    int choice;

    do {

        cout << "\nCustomer Account Management System"<<endl;

        cout << "1. Add Customer"<<endl;

        cout << "2. Edit Customer"<<endl;

        cout << "3. Display All Customers"<<endl;

        cout << "4. Exit"<<endl;

        cout << "Enter your choice: ";

        cin >> choice;

        switch (choice) {

            case 1:

                add\_customer(c, count);

                break;

            case 2:

                edit\_customer(c, count);

                break;

            case 3:

                display\_customer(c, count);

                break;

            case 4:

                cout << "Exiting program..."<<endl;

                break;

            default:

                cout << "Invalid choice. Try again."<<endl;

        }

    } while (choice != 4);

    return 0;

}

**A screenshot of a computer program

Description automatically generated**

**QUESTION#7:**

**Write a program that simulates a soft drink machine. The program should use a structure that stores the following data:**

**Drink Name**

**Drink Cost**

**Number of Drinks in Machine**

**The program should create an array of five structures. The elements should be initialized with the**

**following data:**

**Drink Name Cost Number in Machine**

**Cola .75 20**

**Root Beer .75 20**

**Lemon-Lime .75 20**

**Grape Soda .80 20**

**Cream Soda .80 20**

**Each time the program runs, it should enter a loop that performs the following steps: A list of drinks is displayed on the screen. The user should be allowed to either quit the program or pick a drink. If the user selects a drink, he or she will next enter the amount of money that is to be inserted into the drink machine. The program should display the amount of change that would be returned and subtract one from the number of that drink left in the machine. If the user selects a drink that has sold out, a message should be displayed. The loop then repeats. When the user chooses to quit the program it should display the total amount of money the machine earned. Input Validation: When the user enters an amount of money, do not accept negative values or values greater than $1.00.**

#include<iostream>

using namespace std;

struct drink{

   string name;

   float cost;

   int numDrink;

};

int main(){

    int opt;

    float money=0; int total=0; string name; int numDrinks; float change=0;

 drink d[5] = {

    {"Cola", 0.75, 20},

    {"Root-Beer", 0.75, 20},

    {"Lemon-Lime", 0.75, 20},

    {"Grape-Soda", 0.80, 20},

    {"Cream-Soda", 0.80, 20}

};

do{

  int flag=0;

  cout<<endl<<"------Welcome------"<<endl;

   for(int i=0;i<5;i++){

        cout<<d[i].name<<" "<<d[i].cost<<" "<<d[i].numDrink<<endl;

    }

    cout<<endl<<"1. Buy Drink"<<endl;

    cout<<"2. Quit"<<endl<<endl;

   cout<<"Enter option : ";

   cin>>opt;

   switch(opt){

    case 1: {

        cout<<"Enter Drink Name :";

        cin>>name;

      for(int i=0;i<5;i++){

          if(name==d[i].name){

            flag=1;

            if(d[i].numDrink > 0){

            cout<<"Enter money :";

            cin>>money;

            if(money >= d[i].cost && money <= 1){

                 change=money-d[i].cost;

                 total+=d[i].cost;

                 cout<<"your change is :"<<change<<endl;

                  d[i].numDrink**-=1;**

            }

            else{

                cout<<"Invalid money or enter money less than 1 dollar"<<endl;

                break;

            }

            }

            else{

                cout<<d[i].name<<" is all out"<<endl;

                break;

            }

          }

          if(flag==0){

            cout<<"Invalid name"<<endl;

            break;

          }

      }

      break;

    }

case 2: {

        cout<<"Program Ending...."<<endl;

        return 0;

      }

      default :{

        cout<<"Invalid option";

      }

    }

}while(opt!=2);

}

**RESULT**:

**A screenshot of a computer

Description automatically generated**