Lab 5 answers

//

Pointers :

11.2) Print the consonants

```
#include <iostream>

using namespace std;

int main()
{
   int x,counter=0;
   char y[50000],input;
   cout << "please type the size of the array " ;
   cin >> x;
   for (int i=0;i<x;i++){

      cin >> input;
      if(input!='a' && input!='o' && input!='u' && input!='i' && input!='e'){

      y[counter]= input;
      counter++;
      }
   }

   for (int j=0;j<=counter;j++){

      cout << y[j] << " ";
   }
   return 0;
}
```

//

11.3) Sort an Array

```
#include <iostream>

using namespace std;

void swapelements(int *p1, int* p2){

    int intermed = *p1;
        *p1 = *p2 ;
        *p2 = intermed ;

}


int main()
{   int arr[6] ={1,6,4,3,7,2};
    bool changed = true;
    do
    {
    changed = false;
    for (int i = 0; i < 5 ; i++)
    if (arr[i] > arr[i + 1])
    {
    swapelements(&arr[i], &arr[i + 1]);
    changed = true;
    }
    } while (changed);

    for (int i=0; i<6;i++){


        cout << arr[i] << " ";
    }
    return 0;
}
```

//

11.4) Sum of even locations

```
#include <iostream>

using namespace std;
```

```
int sumOfEven(const int* arri, int z){
    int sum=0 ;
    for (int i=0;i<z;i+=2){
        sum+=arri[i];

    }
return sum;
};

double sumOfEven(const double* arri, int z){
double sum=0 ;
    for (int i=0;i<z;i+=2){
        sum+=arri[i];

    }
return sum;
};

int main()
{
    int arr1[7]={1,2,3,4,5,6,7};
    double arr2[7]={1.1,2.2,3.3,4.4,5.5,6.6,7.7};
    int x= sumOfEven(arr1, 7);
    double y= sumOfEven(arr2, 7);

    cout << x << " " << y <<" " << endl;
    return 0;
}
```

//

11.5) Find the largest element

```
#include <iostream>

using namespace std;

int findthelargest(const int* x, int z){
    int large =0 ;
    for(int i=0;i<z;i++){
        if (x[i] > large ) large = x[i] ;
```

```
    }
    return large;
};


int main()
{
  int x[8]= {6, 7, 9, 10, 15, 3, 99, -21} ;
  int y= findthelargest(x,8) ;
  cout << y << endl;
    return 0;
}
```

//

11.11) Geometry: find the bounding rectangle:

Will have another implementation using classes.

```
#include <iostream>

using namespace std;

double findthemax(const double* x, int z){
    double large =0 ;
    for(int i=0;i<z;i++){
        if (x[i] > large ) large = x[i] ;


    }
    return large;
};

double findthemin(const double* x, int z){
    double minn =100000 ;
    for(int i=0; i<z; i++){
        if (x[i] < minn ) minn = x[i] ;


    }
```

```
    return minn;
};



int main()
{
    cout << "Enter Five Points " << endl;
    double x[5],y[5];
    for (int i=0;i< 5;i++){

        cin >> x[i] >> y[i] ;

    }
    double centre[2]  ;
    double width = findthemax(x,5)- findthemin(x,5);
    double hight = findthemax(y,5)- findthemin(y,5);

    centre[0]= (findthemax(x,5)+findthemin(x,5))/2;
    centre[1]= (findthemax(y,5)+findthemin(y,5))/2;

    cout << "The bounding rectangle's center ("<<centre[0] <<", " << centre[1] << "), width: " <<
width << ", hight: " << hight << endl ;
    return 0;
}
```

//

Structures :

11.1) Movie Data

```
#include <iostream>

using namespace std;

struct MovieData{

    string title;
    string director;
    int year_released;
    int running_time ; // in minutes
```

```
};

void display(MovieData x){

  cout << x.title << " " << x.director << " "  << x.year_released << " " << x.running_time << endl;

}
int main()
{
    MovieData x1 = { "movietitle1", "director1", 1, 30};
    MovieData x2 = {"movietitle2", "director2",2 ,60};
    display(x1);
    display(x2);


    return 0;
}
```

//

11.4) Weather Statistics

```
#include <iostream>

using namespace std;

struct weatherthismonth{

   int TotalRainfall ;
   float HighTemperature ;
   float LowTemperature ;
   float AverageTemperature ;

};


int main()
{
    float avrain=0, avtemp=0, highest[2]={0,0}, lowest[2]={0,0};
    weatherthismonth weather[12];

    for (int i=0; i<12; i++){
```

```
        cout << "please write info of month :"<< i+1 <<endl;
        cout << "TotalRainfall: " ;
        cin >> weather[i].TotalRainfall ;
        cout << "HighTemperature: " ;
        cin >> weather[i].HighTemperature ;
        if (weather[i].HighTemperature > 140 ||  weather[i].HighTemperature < -100){
            cout << "\n this temperature is invalid, it should be between -100 & 140" << endl ;
            i-- ;

            continue ;
        }
        else if(weather[i].HighTemperature > highest[0]){
            highest[0] = weather[i].HighTemperature;
            highest[1]= i+1 ;

        }
        else if(weather[i].HighTemperature < lowest[0]){
            lowest[0] = weather[i].HighTemperature;
            lowest[1]= i+1 ;

        }
        cout << "LowTemperature: " ;
        cin >> weather[i].LowTemperature;
        if (weather[i].LowTemperature > 140 ||  weather[i].LowTemperature < -100){
            cout << "this temperature is invalid, it should be between -100 & 140" << endl ;
            i-- ;

            continue ;
        }
        else if(weather[i].LowTemperature > highest[0]){
            highest[0] = weather[i].LowTemperature;
            highest[1]= i+1 ;

        }
        else if(weather[i].LowTemperature < lowest[0]){
            lowest[0] = weather[i].LowTemperature;
            lowest[1]= i+1 ;

        }
        weather[i].AverageTemperature = (weather[i].HighTemperature +
weather[i].LowTemperature)/2 ;
        avrain += weather[i].TotalRainfall ;
        avtemp += weather[i].AverageTemperature;
```

```
    }

    cout << "average monthly rainfall: " << avrain/12  <<endl;
    cout << "total rainfall for the year: " << avrain << endl;
    cout << "average temperature: " << avtemp/12 << endl;
    cout << "the highest temperature this year was: " << highest[0] << " which was during"<<
highest[1]<< endl;
    cout << "the lowest temperature this year was: " << lowest[0] << " which was during"<<
lowest[1]<< endl;


    return 0;
}
```

//

11.5) Weather Statistics Modification

Only add this to the previous code.

```
enum month { January, February, March, April, May, June, July, August, September, October,
November, December};

    for ( int i=January; i<=December; i++){
```

//

11.9) Speakers' Bureau & 11.7) Customer Accounts same idea like 11.4 .

//

11.11) Monthly Budget

```
#include <iostream>

using namespace std;

struct MonthlyBudget{
```

```
    float Housing ;
    float Utilities ;
    float HouseholdExpenses ;
    float Transportation ;
    float Food ;
    float Medical;
    float Insurance ;
    float Entertainment ;
    float Clothing ;
    float Miscellaneous ;

};


MonthlyBudget getdata(){

    MonthlyBudget x;
      cout << "Please floater the Budget Of this Month :" << endl;
      cout << "Housing : " ;
      cin >> x.Housing ;
      cout << "Utilities : " ;
      cin >> x.Utilities ;
      cout << "Household Expenses : ";
      cin >> x.HouseholdExpenses ;
      cout  << "Transportation : ";
      cin >> x.Transportation ;
      cout << "Food : " ;
      cin >> x.Food ;
      cout << "Medical : " ;
      cin >> x.Medical ;
      cout << "Insurance : " ;
      cin >> x.Insurance ;
      cout << "Entertainment : " ;
      cin >> x.Entertainment ;
      cout << "Clothing : " ;
      cin >> x.Clothing ;
      cout << "Miscellaneous : ";
      cin >> x.Miscellaneous ;



    return x;
}
```

```
void display(MonthlyBudget y){

    // you can add here any functionality or calculations you want .

    cout << "\n This is your Budget \n" << endl;

    cout << "////////////////////////////" << endl;

    cout << " Housing :" << y.Housing << endl <<  " Utilities :" << y.Utilities << endl <<"
HouseholdExpenses :"  ;
    cout << y.HouseholdExpenses << endl <<" Transportation :" << y.Transportation << endl <<"
Food :" << y.Food ;
    cout << endl <<" Medical :" << y.Medical << endl <<" Insurance :" << y.Insurance << endl <<"
Entertainment :" << y.Entertainment ;
    cout<< endl <<" Clothing :" << y.Clothing << endl <<" Miscellaneous :" << y.Miscellaneous <<
endl ;

    float spendings =
y.Clothing+y.Entertainment+y.Food+y.HouseholdExpenses+y.Housing+y.Insurance+y.Medical+
y.Miscellaneous+y.Transportation+y.Utilities ;
    cout << "You spent " << spendings << " $ this month. " << endl;


}


int main()
{
    MonthlyBudget thismonth;
    thismonth = getdata();
    display(thismonth) ;

    return 0;
}
```