6a

#include <iostream>

using namespace std;

int main()

{

const int SIZE=10;

int numbers[SIZE];

cout<<"please enter 10 numbers"<<endl;

for(int count=0;count<SIZE;count++)

{

cin>>numbers[count];

}

int large=0;

int small=9999999;

for(int i=0;i<SIZE;i++)

{

if (numbers[i]<small)

small=numbers[i];

if (numbers[i]>large)

large=numbers[i];

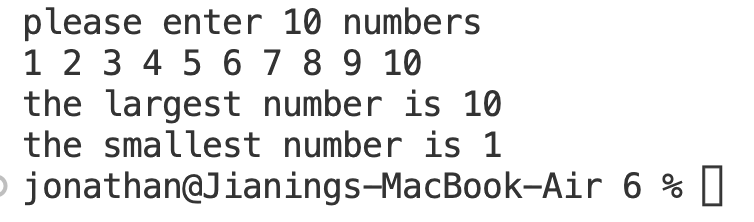
}

cout<<"the largest number is "<<large<<endl;

cout<<"the smallest number is "<<small<<endl;

return 0;

}



6b

#include <iostream>

#include <vector>

using namespace std;

const int ARRAY\_SIZE = 12;

double input(double);

void getValues(double []);

double average(double);

double calculateTotal(double []);

void findHighestMonths(double [], vector<int> &);

void findLowestMonths(double [], vector<int> &);

void display(double, double, vector<int>, vector<int>);

int main()

{

double rainfall[ARRAY\_SIZE];

cout << "enter rainfall each month: " << endl;

getValues(rainfall);

double total\_rainfall = calculateTotal(rainfall);

double avarage\_rainfall = average(total\_rainfall);

vector<int>highest\_months;

findHighestMonths(rainfall, highest\_months);

vector<int>lowest\_months;

findLowestMonths(rainfall, lowest\_months);

display(total\_rainfall, avarage\_rainfall, highest\_months, lowest\_months);

return 0;

}

double input(double user\_number)

{

while(!(cin >> user\_number) || user\_number < 0)

{

cout << "Error: please enter a positive number."

<< "Try again: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

return user\_number;

}

void getValues(double array[])

{

for (int i = 0; i < ARRAY\_SIZE; i++)

{

cout << (i + 1)

<< ": ";

array[i] = input(array[i]);

}

}

double calculateTotal(double array[])

{

double total = 0;

for (int i = 0; i < ARRAY\_SIZE; i++)

total += array[i];

return total;

}

double average(double sum)

{

return sum / ARRAY\_SIZE;

}

void findHighestMonths(double array[], vector<int> &vector\_array)

{

double max = array[0];

vector\_array.push\_back(0);

for (int i = 1; i < ARRAY\_SIZE; i++)

{

if(array[i] > max)

{

max = array[i];

vector\_array.clear();

vector\_array.push\_back(i);

}

else if (array[i] == max)

{

max = array[i];

vector\_array.push\_back(i);

}

}

}

void findLowestMonths(double array[], vector<int> &vector\_array)

{

double min = array[0];

vector\_array.push\_back(0);

for (int i = 1; i < ARRAY\_SIZE; i++)

{

if(array[i] < min)

{

min = array[i];

vector\_array.clear();

vector\_array.push\_back(i);

}

else if (array[i] == min)

{

min = array[i];

vector\_array.push\_back(i);

}

}

}

void display(double total,

double average,

vector<int> highest\_months,

vector<int> lowest\_months)

{

const string MONTHS[] = {"January", "February", "March",

"April" , "May" , "June",

"July" , "August" , "September",

"October", "November", "December"};

cout << "total= "

<< total

<< endl;

cout << "average= "

<< average

<< endl;

cout << "highest= ";

for (int i = 0; i < highest\_months.size(); i++)

{

if(i == highest\_months.size() - 1)

cout << MONTHS[highest\_months[i]];

else

cout << MONTHS[highest\_months[i]] << ", ";

}

cout << endl;

cout << "lowest= ";

for (int i = 0; i < lowest\_months.size(); i++)

{

if(i == lowest\_months.size() - 1)

cout << MONTHS[lowest\_months[i]];

else

cout << MONTHS[lowest\_months[i]] << ", ";

}

cout << endl;

}

Text

Description automatically generated with low confidence

6c

#include <iostream>

using namespace std;

const int SIZE = 5;

void getdata(const string[], int[]);

void findBest\_WorstSeller(const int[], int&,int&);

int findTotal(const int[]);

void printSummary(const string[], const int[], int, int, int);

int main()

{

string salsa[SIZE]= {"hot", "zesty", "mild", "medium", "sweet"};

int jarsSold[SIZE], totalSale;

int bestIndex = 0, worstIndex = 0;

getdata(salsa,jarsSold);

totalSale = findTotal(jarsSold);

findBest\_WorstSeller(jarsSold,bestIndex, worstIndex );

printSummary (salsa, jarsSold,totalSale, bestIndex, worstIndex);

return 0;

}

void getdata(const string s[], int j[])

{

for (int i = 0; i < SIZE; i++)

{

cout << "How many jars of " << s[i]<< " salsa were sold ::: ";

cin >> j[i];

while(j[i]< 0)

{

cout << " Please enter a positive number ::: ";

cin >> j[i];

}

}

}

int findTotal(const int j[])

{

int total = 0;

for(int i = 0; i < SIZE; i++)

total = total + j[i];

return total;

}

void findBest\_WorstSeller(const int j[], int& b, int& w)

{

int best = j[0] ;

int worst = j[0];

for(int i = 0; i < SIZE; i++)

{

if(best < j[i])

{

best = j[i];

b = i;

}

if(worst > j[i])

{

worst = j[i];

w = i;

}

}

}

void printSummary(const string s[], const int j[], int t, int b, int w)

{

cout << "\n\n S U M M A R Y \n\n";

cout << " Salsa sold : \n";

for( int i = 0; i < SIZE; i++)

{

cout << s[i]<< " " << j[i] << endl;

}

cout <<"\n\nTotal sales " << t << endl << endl;

cout <<"Best seller " << s[b] << " " << j[b] << " jars sold\n\n";

cout <<"Worst seller " << s[w] << " " << j[w] << " jars sold\n\n";

}

A picture containing text

Description automatically generated