Fundamentals of Programming

Lab Manual 10

Name: Mehwish Nadeem

CMS ID: 455426

**Task 1:**

#include <iostream>

#include <vector>

using namespace std;

int main(){

vector<int> v;

vector<int>::iterator rem=v.begin()+5;

for(int i=1; i<11; i++){

v.push\_back(i);

}

cout<<"The values in the vector are: \n";

for(vector<int>::iterator i=v.begin(); i!=v.end(); i++){

cout<<\*i<<'\t';

}

cout<<endl;

v.push\_back(5);

rem=v.begin()+5;

v.erase(rem);

cout<<"After pushing 5 and removing integer at posiiton 5"<<endl;

for(vector<int>::iterator i=v.begin(); i!=v.end(); i++){

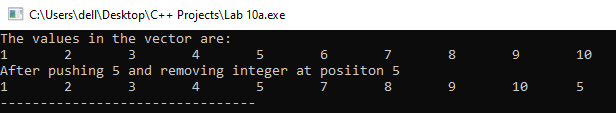
cout<<\*i<<'\t';

}

return 0;

}

**Task 1 Output:**

****

**Task 2:**

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int main(){

vector<string> name;

string iname, temp\_name;

vector<int> grade;

vector<int> mode\_score;

int igrade, sum=0, mean, temp=0, temp\_grade;

cout<<"Input the name and grade of each student."<<endl

<<"Input Backslash 0 in the name and a charecter in the grade to stop inputs."<<endl;

while(cin>>iname && cin>>igrade){

name.push\_back(iname);

grade.push\_back(igrade);

}

cout<<"The values in the array are: \n";

for(int i=0; i<name.size(); i++){

cout<<name[i]<<"\t"<<grade[i]<<endl;

}

for(int i=0; i<grade.size(); i++){

sum+=grade[i];

}

for(int i=0; i<grade.size(); i++){

mode\_score.push\_back(0);

}

for(int i=0; i<grade.size(); i++){

for(int j=1; j<grade.size(); j++){

if(grade[i]>grade[j]){

temp\_grade=grade[i];

grade[i]=grade[j];

grade[j]=temp\_grade;

temp\_name=name[i];

name[i]=name[j];

name[j]=temp\_name;

}

}

}

cout<<"The median score is: ";

if(grade.size()%2==0){

cout<<endl

<<name[name.size()/2]<<'\t'<<grade[grade.size()/2]<<endl;

}

else{

cout<<grade[((grade.size()/2)+((grade.size()/2)-1))/2]<<endl;

}

mode\_score.resize(grade.size());

mean=sum/grade.size();

for(int i=0; i<grade.size(); i++){

for(int j=0; j<grade.size(); j++){

if(grade[i]==grade[j]){

mode\_score[i]++;

}

}

}

for(int i=0; i<mode\_score.size()-1; i++){

if(mode\_score[i]>temp){

temp=mode\_score[i];

}

}

cout<<"The mean of the grade is "<<mean<<endl;

cout<<"The most repeated score and scorer is/are: \n";

for(int i=0; i<grade.size(); i++){

if(mode\_score[i]==temp){

cout<<name[i]<<'\t'<<grade[i]<<endl;

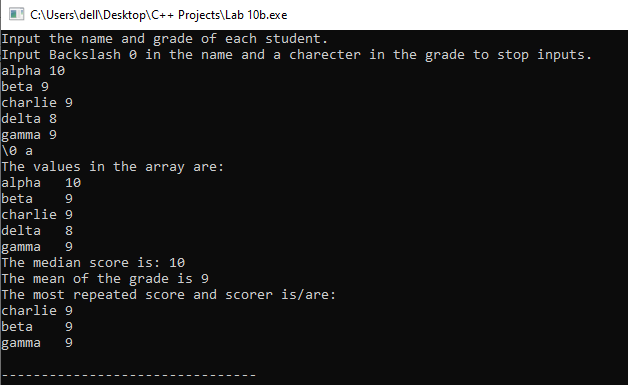
}

}

return 0;

}

**Task 2 Output:**

****

**Task 3:**

#include <bits/stdc++.h>

using namespace std;

class Triangle{

public:

float dimension1, dimension2, dimension3;

float area(){

return (dimension1\*dimension2)/2;

}

float perimeter(){

return dimension1+dimension2+dimension3;

}

};

int main(){

Triangle test\_1;

cout<<"Input the dimesntions of the triangle. \n"

<<"Base: ";

cin>>test\_1.dimension1;

cout<<"Height: ";

cin>>test\_1.dimension2;

cout<<"Hypotenuse: ";

cin>>test\_1.dimension3;

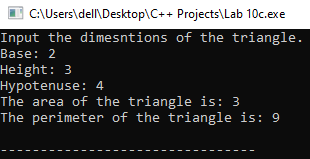
cout<<"The area of the triangle is: "<<test\_1.area()<<endl

<<"The perimeter of the triangle is: "<<test\_1.perimeter()<<endl;

return 0;

}

**Task 3 Output:**

****

**Task 4:**

#include <bits/stdc++.h>

using namespace std;

struct salary\_table{

string name;

int base\_pay, work\_hours, salary;

void pay\_inc(){

if(work\_hours<8){

salary=base\_pay;

}

else if(work\_hours<10 && work\_hours>=8){

salary=base\_pay+50;

}

else if(work\_hours<12 && work\_hours>=10){

salary=base\_pay+100;

}

else{

salary=base\_pay+150;

}

}

};

int main(){

int num, check;

cout<<"Input the number of employees: ";

cin>>num;

salary\_table employee[num];

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

cout<<"Input the Name, Base Pay and Hours of Work Per Day of Employee "<<i+1<<endl;

cin>>employee[i].name;

cin>>employee[i].base\_pay;

cin>>employee[i].work\_hours;

employee[i].pay\_inc();

}

cout<<"Input the enrty number to check employe salary: \n"

<<"Input a charecter to exit \n";

do{

cin>>check;

if(check>=1 && check<=num){

cout<<employee[check-1].name<<'\t'

<<employee[check-1].work\_hours<<'\t'

<<employee[check-1].salary<<'\n';

}

else{

cout<<"Enrty number does not exist."<<endl;

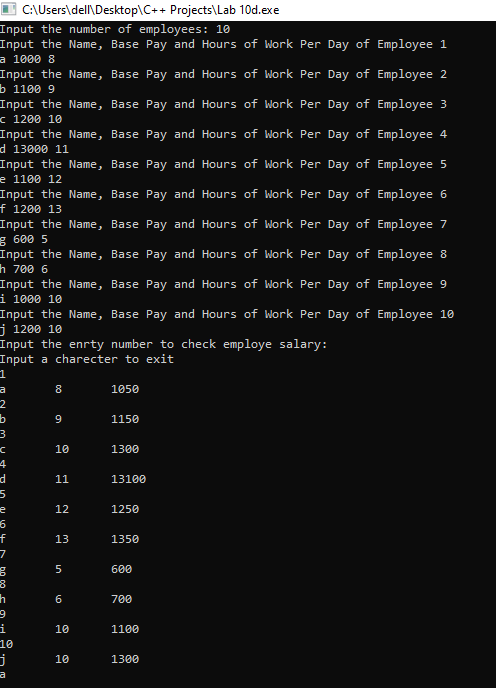
}

} while(!cin.fail());

return 0;

}

**Task 4 Output:**

****