Name- Gurkirat Singh
Section-C
Roll no. - 832025
Assignment-7 on Structures
Due Date: March 12, 2021

Note: For Q1 and Q2 use . operator, For Q3,Q4,Q5 use -> operator to access structure members.

Use dot (.) operator to access structure members.

1) A railway employee is paid 1200/- (rupees) per day for regular 8 hours of work. Any hours over that are paid overtime rate of 100/- per hour. From the employee's gross pay (total pay per month), 2% is deducted for professional tax, 10% for provident fund and 5% for income tax. However, the employee will get 2% (of the gross pay) for the education of a child. Write a program to create a structure of employee and read data of 'N' employees as follows:
i)the number of extra hours (which the employee worked during a month)
ii)the number of children the employee has.
The program should output the ' N' employees gross pay (total pay earned by the employee by working) and net take-home pay (after deductions and earning for child education). Assume all months have 30 days.

Ans.

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

struct employee{
	float extra_hours;
	int num_of_children;
};

int main(void){
	int num_of_employees;
	float perday_salary = 1200;
	int days = 30;
	float overtime_per_hour = 100;
	float professional_tax = 2;
	float provident_fund = 10;
	float income_tax = 5;
	float child_support = 2;
	cout<<"enter number of employees"<<endl;
	cin>>num_of_employees;
	employee array[num_of_employees];

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

```
                cout<<"Enter number of extra hours worked and number of children as space
separated numbers"<<endl;
                cin>>array[i].extra_hours>>array[i].num_of_children;
        }
        for(int i = 0; i<num_of_employees; i++){
                float gross_salary_per_month = perday_salary * days +
(array[i].extra_hours*overtime_per_hour);
                float take_home = gross_salary_per_month +
(array[i].num_of_children*child_support*gross_salary_per_month/100) -
(provident_fund*gross_salary_per_month/100) -
(professional_tax*gross_salary_per_month/100) -
(income_tax*gross_salary_per_month/100);
                cout<<"Gross monthly salary of employee "<<i+1<<" is:
"<<gross_salary_per_month<<endl;
                cout<<"net take-home monthly salary of employee "<<i+1<<" is:
"<<take_home<<endl;
        }
        return 0;
}
```

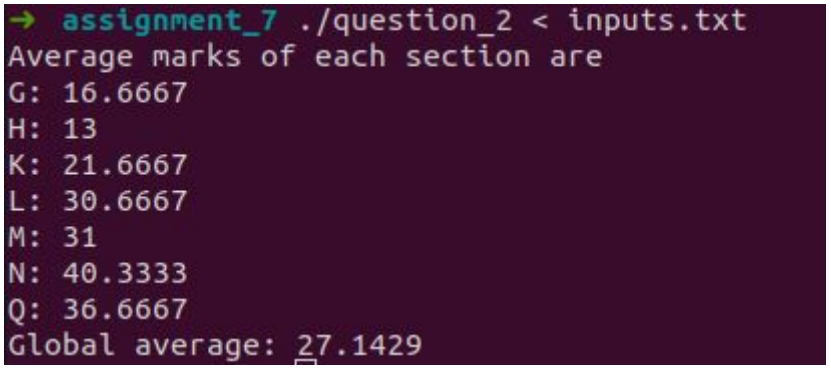
```
→ assignment_7 ./question_1
enter number of employees
5
Enter number of extra hours worked and number of children as space separated numbers
5 4
Enter number of extra hours worked and number of children as space separated numbers
2 3
Enter number of extra hours worked and number of children as space separated numbers
6
1
Enter number of extra hours worked and number of children as space separated numbers
9 2
Enter number of extra hours worked and number of children as space separated numbers
1 1
Gross monthly salary of employee 1 is: 36500
net take-home monthly salary of employee 1 is: 33215
Gross monthly salary of employee 2 is: 36200
net take-home monthly salary of employee 2 is: 32218
Gross monthly salary of employee 3 is: 36600
net take-home monthly salary of employee 3 is: 31110
Gross monthly salary of employee 4 is: 36900
net take-home monthly salary of employee 4 is: 32103
Gross monthly salary of employee 5 is: 36100
net take-home monthly salary of employee 5 is: 30685
```

2) Consider a structure with the members: Roll_No, Section, Marks_in_PSCP. Write a program to read the details of six sections (G, H, K, L, M, N and Q) students where each section contains 80 students. Write a program to calculate the average marks for each section and also the average of all the seven sections.

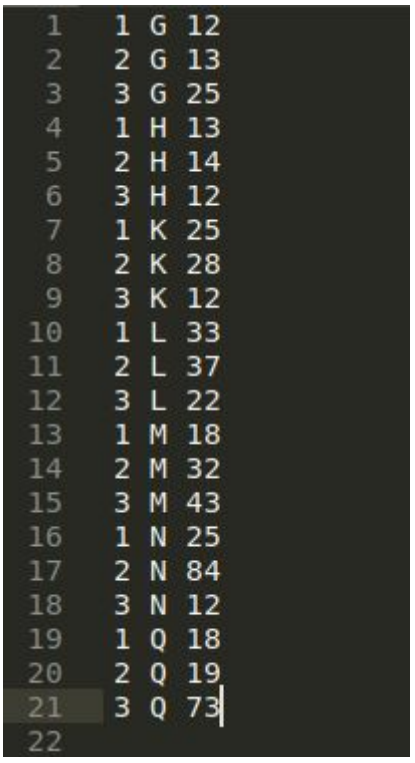
Ans.

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

struct student{
	int roll_no;
	char section;
	float marks_in_PSCP;
};

struct avg_marks{
	char section;
	float avrg_marks;
};

int main(void){
	int num_students_each_sec = 3;
	int num_sec = 7;
	int total_students = num_students_each_sec*num_sec;
	student array[num_students_each_sec*num_sec]{};
	char array_sections[num_sec] = {'G', 'H', 'K', 'L', 'M', 'N', 'Q'};
	avg_marks section_array[num_sec];
	float global_avg;
	float total_sum = 0;
	for(int i = 0; i<num_students_each_sec*num_sec; i++){
		cin>>array[i].roll_no>>array[i].section>>array[i].marks_in_PSCP;
		total_sum += array[i].marks_in_PSCP;
	}
	global_avg = total_sum/total_students;


	for(int i = 0; i<num_sec; i++){
		section_array[i].section = array_sections[i];

	}

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

```
            float sum = 0;
            for(int j = 0; j<num_students_each_sec*num_sec; j++){
                if(array[j].section == section_array[i].section){

                    sum += array[j].marks_in_PSCP;
                }
            }
            section_array[i].avrg_marks = sum/num_students_each_sec;
        }
        cout<<"Average marks of each section are"<<endl;

        for(int i = 0; i<num_sec; i++){
            cout<<section_array[i].section<<": "<<section_array[i].avrg_marks<<endl;
        }
        cout<<"Global average: "<<global_avg<<endl;



        return 0;
}
```



Output



inputs used(written in a .txt file)

3) Write a program to create a structure ‘student’ with the member variable number, name, marks and branch. Read sixty students details. Then your program should display the names of the students who got more than 60 marks of CSE branch with name ‘Aditya’.

Ans.

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

struct student{
	int number;
	string name;
	float marks;
	string branch;
};

int main(void){
	int num_students = 7;
	student array[num_students];
	student* ptr_array = array;
	for(int i = 0; i<num_students; i++){
		cin>>(ptr_array+i) -> number>>(ptr_array+i) -> name>>(ptr_array+i) ->
marks>>(ptr_array+i) -> branch;
	}
	for(int i = 0; i<num_students; i++){
		if((ptr_array+i) -> name == "Aditya" && (ptr_array+i) -> branch == "CSE" &&
(ptr_array+i) -> marks>60){
			cout<<(ptr_array+i)->number<<" | ";
			cout<<(ptr_array+i)->name<<" | ";
			cout<<(ptr_array+i)->marks<<" | ";
			cout<<(ptr_array+i)->branch<<endl;
		}
	}
	return 0;
}
```

```
➜ assignment_7 ./question_3 < inputs_3.txt
2 | Aditya | 64 | CSE
5 | Aditya | 77 | CSE
```

Output

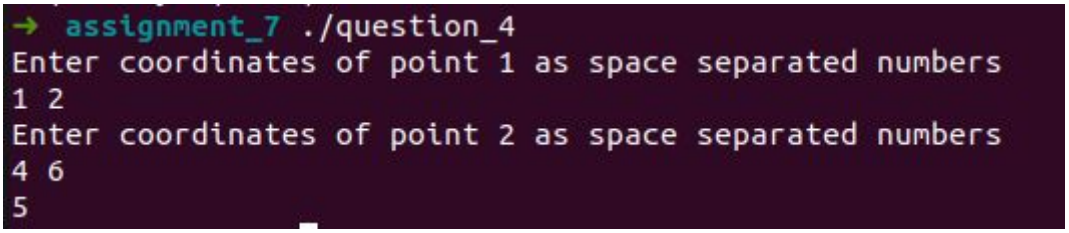
```
1 Gurkirat 79 CSE
2 Aditya 64 CSE
3 Aditya 32 CSE
4 Aditya 66 ECE
5 Aditya 77 CSE
6 Adarsh 77 CSE
7 TNA 88 BIOTECH
```

input from .txt file

4) A Point on the 2-D plane can be represented by two numbers: an X-coordinate and Y-Coordinate. For example, (2, 3) represents a point 2 units to the right of the origin along the x-axis and 3-units up the y-axis. Write a program that uses a structure called Point to model a point. Define two points, the user have to input values of two points. The Program should calculate the Euclidean distance between the two input points and display it.

Ans.

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

struct point{
	float x;
	float y;
};

int main(void){
	point p1;
	point *pointer_p1;
	pointer_p1 = &p1;
	point p2;
	point *pointer_p2;
	pointer_p2 = &p2;
	cout<<"Enter coordinates of point 1 as space separated numbers"<<endl;
	cin>>(pointer_p1 -> x)>>(pointer_p1 -> y);
	cout<<"Enter coordinates of point 2 as space separated numbers"<<endl;
	cin>>(pointer_p2 -> x)>>(pointer_p2 -> y);
	float distance = pow(pow((pointer_p2 -> x)-(pointer_p1 -> x), 2) + pow((pointer_p2 ->
y)-(pointer_p1 -> y), 2), 0.5);
	cout<<distance<<endl;
	return 0;
}
```



5) Write a program to create a structure called BankDeposit with members amoun (amount to deposit in bank), tenure (No. of years deposit to be maintained). Create another structure called Dates with members date, month, year (date, month, year of int type). The program should read DOB of a person, and Date of deposit using Dates structure variables and calculate present age of the person. If the person is senior citizen (age >=60 yrs) then rate of interest is 9% else 8%.
Calculate the total amount that person receives after date of maturity (date of deposit+ tenure).

Ans.

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

struct BankDeposit{
	float deposit_in_bank;
	int tenure;
};

struct Dates{
	int date;
	int month;
	int year;
};

int main(void){

	Dates dob;
	Dates* ptr_dob;
	ptr_dob = &dob;

	Dates deposits;
	Dates* ptr_deposits;
	ptr_deposits = &deposits;

	BankDeposit first_deposit;
	BankDeposit* ptr_first_deposit;
	ptr_first_deposit = &first_deposit;

	cout<<"Enter date of birth in form of DD MM YYYY"<<endl;
	cin>>ptr_dob -> date>>ptr_dob -> month>>ptr_dob -> year;

	cout<<"Enter date of deposit in form of DD MM YYYY"<<endl;
	cin>>ptr_deposits -> date>>ptr_deposits -> month>>ptr_deposits -> year;
```

```
	cout<<"Enter amount"<<endl;
	cin>>ptr_first_deposit -> deposit_in_bank;
	cout<<"Enter tenure"<<endl;
	cin>>ptr_first_deposit -> tenure;

	float interest_rate;
	int age;

	if(ptr_deposits -> year - ptr_dob -> year > 60){
		interest_rate = 9;
		age = ptr_deposits -> year - ptr_dob -> year;
	}
	else if(ptr_deposits -> year - ptr_dob -> year == 60){
		if(ptr_deposits -> month > ptr_dob -> month){
			interest_rate = 9;
			age = ptr_deposits -> year - ptr_dob -> year;
		}
		else if(ptr_deposits -> month < ptr_dob -> month){
			interest_rate = 8;
			age = ptr_deposits -> year - ptr_dob -> year - 1;
		}
		else{
			if(ptr_deposits -> date > ptr_dob -> date){
				interest_rate = 9;
				age = ptr_deposits -> year - ptr_dob -> year;
			}
			else if(ptr_deposits -> date < ptr_dob -> date){
				interest_rate = 8;
				age = ptr_deposits -> year - ptr_dob -> year - 1;
			}
			else{
				interest_rate = 9;
				age = ptr_deposits -> year - ptr_dob -> year;
			}
		}

	}
	else{
		interest_rate = 8;
		age = ptr_deposits -> year - ptr_dob -> year;
	}

	double amount = (ptr_first_deposit -> deposit_in_bank)*pow(1+interest_rate/100,
ptr_first_deposit -> tenure);
	cout<<"Age: "<<age<<endl;
	cout<<"interest_rate: "<<interest_rate<<endl;
	cout<<"Amount: "<<amount<<endl;
```

```
    return 0;

}
```

```
➜ assignment_7 ./question_5
Enter date of birth in form of DD MM YYYY
1 7 1957
Enter date of deposit in form of DD MM YYYY
11 08 2021
Enter amount
1000
Enter tenure
5
Age: 64
interest_rate: 9
Amount: 1538.62
```