Computer Programming

Assignment Semester 1

Kushagra Lakhwani
2021UCI8036

Contents

1 Questions

1.1 Student Grades

1.1.1 Statement

Write a program in C that uses a two-dimensional array to store the numeric grade for each student (n) in a multiple teacher's class (m). The program assumes that the teacher has three classes and a maximum of 30 students per class. Both the variable M and N should be user defined.

1.1.2 Code

```c
#include <stdio.h>

struct student
{
  char grade;
};

void get_marks(student classes[][30])
{
  int m, n;

  printf("\nEnter class: ");
  scanf("%d", &m);
  printf("Enter student: ");
  scanf("%d", &n);

  printf("The given student has got %c grade!\n", classes[m - 1][n - 1]);
}

void populate_classes(student classes[3][30])
{
  int num;

  for (int i = 0; i < 3; i++)
  {
    student *curr_class = classes[i];
    printf("Enter number of students in class %d: ", i + 1);
    scanf("%d", &num);

    for (int j = 0; j < num; j++)
    {
      printf("Enter the Grade of student %d in class %d: ", j + 1, i + 1);
      scanf(" %c", &(curr_class[j].grade)); // blank space is important
    }
    printf("\n");
```

```
  }
}

int main(int argc, char const *argv[])
{

  student classes[3][30];

  populate_classes(classes);
  get_marks(classes);

  return 0;
}
```

1.1.3 Output

```
➜ Code ./student_grades
Enter number of students in class 1: 3
Enter the Grade of student 1 in class 1: A
Enter the Grade of student 2 in class 1: B
Enter the Grade of student 3 in class 1: C

Enter number of students in class 2: 2
Enter the Grade of student 1 in class 2: A
Enter the Grade of student 2 in class 2: B

Enter number of students in class 3: 2
Enter the Grade of student 1 in class 3: F
Enter the Grade of student 2 in class 3: A


Enter class: 2
Enter student: 1
The given student has got A grade!
➜ Code
```

1.2 Average Age

1.2.1 Statement

Write the program to input the value of age of employees in the company. You have to calculate the average age of the employee in the company using pointer of array.

1.2.2 Code

```c
#include <stdio.h>
#include <stdlib.h>

float avg(int arr[], int n)
{
  float sum = 0;
  for (int i = 0; i < n; i++)
    sum += arr[i];
  return sum / (float)n;
}

int main(int argc, char const *argv[])
{
  int n;
  printf("Enter number of employees: ");
  scanf("%d", &n);

  int *employees = (int *)malloc(n * sizeof(int));

  for (int i = 0; i < n; i++)
  {
    printf("Enter age of employee %d: ", i + 1);
    scanf("%d", &employees[i]);
  }

  printf("The average age of employees is = %.2f\n", avg(employees, n));

  free(employees);

  return 0;
}
```

1.2.3 Output

```
➜  Code git:(master) ✗ ./average_ages
Enter number of employees: 3
Enter age of employee 1: 5
Enter age of employee 2: 8
Enter age of employee 3: 16
The average age of employees is = 9.67
➜  Code git:(master) ✗
```

1.3 Length of string

1.3.1 Statement

A user has given a random size string to input, you have to calculate the length of the string using pointer. You cannot use predefined function strrev.

1.3.2 Code

```c
#include <stdio.h>
int main()
{
  char str[100], i;
  printf("Enter a string: ");
  scanf("%[^\n]s", str);

  // '\0' represents end of String
  for (i = 0; str[i] != '\0'; ++i);
  printf("\nLength of input string: %d\n", i);

  return 0;
}
```

1.3.3 Output

```
➜ Code git:(master) ✗ ./strlen
Enter a string: this is a string that is given to the program

Length of input string: 45
➜ Code git:(master) ✗
```

1.4 Start-Up Owner

1.4.1 Statement

A start-up owner is interested to maintain the dataset of the newly recruited employees.

She is interested in storing the Emp_Name (Str), Emp_Age (int), Emp_Degree (Str), Emp_Exp (Float), Emp_add (Structure). Emp_add needs one user defined data to store street no, city, district and state for the employee address. You have to design a database where we can store all the information for at least 20 employees.

1.4.2 Code

```
#include <stdio.h>

struct address
{
  int street;
  char city[50];
  char district[50];
  char state[50];
};

struct employee
{
  char Emp_Name[50];
  int Emp_Age;
  char Emp_Degree[50];
  float Emp_Exp;
  address Emp_add;
};

void add_emploee(employee &Employee, int i)
{
  printf("\nEnter Name of employee %d: ", i);
  scanf("%s", &Employee.Emp_Name);
  printf("Enter Age of employee %d: ", i);
  scanf("%d", &Employee.Emp_Age);
  printf("Enter Degree of employee %d: ", i);
  scanf("%s", &Employee.Emp_Degree);
  printf("Enter Experience of employee %d: ", i);
  scanf("%f", &Employee.Emp_Exp);

  printf("*Address Details*\n");
  printf("Enter City of employee %d: ", i);
  scanf("%s", &Employee.Emp_add.city);
  printf("Enter District of employee %d: ", i);
  scanf("%s", &Employee.Emp_add.district);
  printf("Enter State of employee %d: ", i);
```

```
  scanf("%s", &Employee.Emp_add.state);
  printf("Enter Street of employee %d: ", i);
  scanf("%d", &Employee.Emp_add.street);
}

void print_employee(employee Employee)
{
  printf("\n%s %s %s", Employee.Emp_Name, Employee.Emp_Degree, Employee.Emp_
}

int main(int argc, char const *argv[])
{
  int n;
  printf("Enter Number of employees: ");
  scanf("%d", &n);

  employee Employees[n];

  for (int i = 0; i < n; i++)
    add_emploee(Employees[i], i + 1);

  for (int i = 0; i < n; i++)
    print_employee(Employees[i]);
  return 0;
}
```

1.4.3 Output

```
➜  Code git:(master) ✗ ./employee
Enter Number of employees: 1

Enter Name of employee 1: John
Enter Age of employee 1: 23
Enter Degree of employee 1: Betch
Enter Experience of employee 1: 2
*Address Details*
Enter City of employee 1: Delhi
Enter District of employee 1: Dwarka
Enter State of employee 1: Delhi
Enter Street of employee 1: 3

John Betch Delhi%
➜  Code git:(master) ✗
```

1.5 Student Names

1.5.1 Statement

Defined a two-dimensional matrix (char)[50][20] to store the student's name in the class. We are expecting to store the 50 students with different length name. Write a program to print all the name with the help of pointers

1.5.2 Code

```c
#include <stdio.h>

void print_names(char students[][20], int n)
{
  printf("\nStudents are: \n");
  for (int i = 0; i < n; i++)
    printf("%s\n", students[i]);
}

int main(int argc, char const *argv[])
{
  int n;
  char students[50][20];
  printf("Enter a number: ");
  scanf("%d", &n);

  for (int i = 0; i < n; i++)
  {
    printf("Enter the name for student %d: ", i + 1);
    scanf("%s", students[i]);
  }
  print_names(students, n);
  return 0;
}
```

1.5.3 Output

```
➜ Code git:(master) ✗ ./students
Enter a number: 4
Enter the name for student 1: John
Enter the name for student 2: KorigamiK
Enter the name for student 3: doe
Enter the name for student 4: aka

Students are:
John
KorigamiK
doe
aka
➜ Code git:(master) ✗
```

1.6 XOR Operation

1.6.1 Statement

The outcome of a XOR operation is true if and only if one operand (but not both) is true. Write a program in 'C' which returns the outcome of an Exclusive OR operation performed on its two operands

1.6.2 Code

```c
#include <stdio.h>

int main(int argc, char const *argv[])
{
  int a, b;
  printf("Enter 2 numbers:\n");
  scanf("%d", &a);
  scanf("%d", &b);
  int xor_result = (a | b) & (~(a & b));
  printf("The XOR of %d and %d is equal to %d\n", a, b, xor_result);
  return 0;
}
```

1.6.3 Output

```
➜  Code git:(master) ✗ ./xor
Enter 2 numbers:
3 5
The XOR of 3 and 5 is equal to 6
➜  Code git:(master) ✗
```

1.7 Left and Right Shift

1.7.1 Statement

Write a program in C to show that Right shift effectively divides a number by 2 and a left shift effectively multiplies a number by 2

1.7.2 Code

```c
#include <stdio.h>

int main(int argc, char const *argv[])
{
  int a;
  printf("Enter a numbers: ");
  scanf("%d", &a);

  for (int i = 0; i < 4; i++)
    printf("%d Right shifted %d times is equal to %d\n", a, i, a >> i);
  printf("\n");
  for (int i = 0; i < 4; i++)
    printf("%d Left shifted %d times is equal to %d\n", a, i, a << i);

  return 0;
}
```

1.7.3 Output

```
➜  Code git:(master) ✗ ./shifts
Enter a numbers: 20
20 Right shifted 0 times is equal to 20
20 Right shifted 1 times is equal to 10
20 Right shifted 2 times is equal to 5
20 Right shifted 3 times is equal to 2

20 Left shifted 0 times is equal to 20
20 Left shifted 1 times is equal to 40
20 Left shifted 2 times is equal to 80
20 Left shifted 3 times is equal to 160
➜  Code git:(master) ✗
```

1.8 Magic Number

1.8.1 Statement

Using the ? Operator, rewrite the magic number program discussed in the class

1.8.2 Code

1.8.3 Output

1.9 Quotient

1.9.1 Statement

Using if else statement write a program in 'C' to read two integers from the user and display the quotient. Your program should be able to detect divide by zero.

1.9.2 Code

```c
#include <stdio.h>

int main(int argc, char const *argv[])
{
  float a, b;
  printf("Enter 2 numbers:\n");
  scanf("%f", &a);
  scanf("%f", &b);
  b == 0
    ? printf("Undefined Behavior\n")
    : printf("The quotient is: %.2f\n", a / b);
  return 0;
}
```

1.9.3 Output

```
➜ Code git:(master) ✗ ./quotient
Enter 2 numbers:
15 25
The quotient is: 0.60
➜ Code git:(master) ✗ ./quotient
Enter 2 numbers:
2 0
Undefined Behavior
➜ Code git:(master) ✗
```

1.10 Text File

1.10.1 Statement

Write a program in C that inputs lines of text until a blank line is entered. Then it redisplays each line one character at a time

1.10.2 Code

1.10.3 Output

1.11 Queue

1.11.1 Statement

Write a program in C using pointers to implement insertion and deletion in a queue. A queue is a data structure that follows a first in first out i.e. the element to go in first is the one to come out first

1.11.2 Code

1.11.3 Output

1.12 Temperature Conversion

1.12.1 Statement

Write a program to print the corresponding celsius to Fahrenheit table. Modify the temperature conversion program to print the table in reverse order, that is from 300 to 0

1.12.2 Code

1.12.3 Output

1.13 Text in File

1.13.1 Statement

Write a program to count blanks, tabs and newlines

1.13.2 Code

1.13.3 Output

1.14 Frequencies

1.14.1 Statement

Write a program to print the histogram of the frequencies of different characters of its input.

1.14.2 Code

1.14.3 Output

References

[1] Author. *Title*. URL: URL.

[2] Article Author. "Article Title". In: *Journal* Volume #.Number # (2021), Pages #. DOI: DOI.

[3] Book Author. *Book Title*. Address: Publisher, 2021.