WEEK-END ASSIGNMENT-03

Operating Systems Workshop (CSE 3541)

Problem Statement:

Working with repetition control structure (**`for`**, **`while`** and **`do-while`**) in programming.

Assignment Objectives:

To learn how to use C repetition control structure in programming and when to use each type in developing programs.

Instruction to Students (If any):

Students are required to write his/her own program by avoiding any kind of copy from any sources. Additionally, They must be able to realise the outcome of that question in relevant to systems programming. You may use additional pages on requirement.

Programming/ Output Based Questions:

1. Find the output/ error of following code snippet

Code snippet: 1(a)

```
int i = 0;
while (i <= 5) {
  printf("%3d %3d\n", i, 10 - i);
  i = i + 1;
 }
```

Output

Code snippet: 1(b)

```
int i=1;
while ( )
{
  printf ( "%d ", i++ ) ;
  if(i>10)
    break ;
}
```

Output

Output

Code snippet: 1(c)

```
int a=1;
do {
  printf("%d ", a++);
 } while ( a < 10 );
```

Code snippet: 1(d)

```
int i, j,n=5;
for(i=1, j=1; j<= n; i+= 2, j++){
  printf("%d%d\n", i, j);
}
```

Output

Code snippet: 1(e)

```
int count = 11;
while (--count+l);
  printf("count down is %d \n",count);
```

Output

Code snippet: 1(e)

```
float x = 1.1 ;
while ( x == 1.1 ) {
        printf ( "%f\n", x ) ;
        x = x - 0.1 ;
}
```

Output

2. During execution of the following program segment, how many lines of hash marks are displayed?

```
for (m = 9; m > 0; --m)
  for (n = 6; n > 1; --n)
    printf("########\n");
```

Output

3. During execution of the following program segment:

   a. How many times does the first call to **`printf`** execute?

   b. How many times does the second call to **`printf`** execute?

   c. What is the last value displayed?

```
for (m = 10; m > 0; --m){
  for (n = m; n > 1; --n)
    printf("%d ", m + n);
  printf("\n");
}
```

Output▼

a.

b.

c.

4. An integer n is divisible by 9 if the sum of its digits is divisible by 9. Develop a program to display each digit, starting with the rightmost digit. Your program should also determine whether or not the number is divisible by 9. Test it on the following numbers:
   n = 154368
   n = 621594
   n = 123456
   Hint: Use the % operator to get each digit; then use / to remove that digit. So 154368 % 10 gives 8 and 154368 / 10 gives 15436. The next digit extracted should be 6, then 3 and so on.

Space for Program and output ▼

5. The greatest common divisor (gcd) of two integers is the product of the integers common factors. Write a program that inputs two numbers and implements the following approach to finding their gcd. We will use the numbers -252 and 735. Working with the numbers' absolute values, we find the remainder of one divide by the other.

$$\begin{array}{r|r} & 0 \\ \cline{2-2} 735 & 252 \\ & -\ \ 0 \\ \cline{2-2} & 252 \end{array}$$

Now we calculate the remainder of the old divisor divided by the remainder found.

$$\begin{array}{r|r} & 2 \\ \cline{2-2} 252 & 735 \\ & -\ \ 504 \\ \cline{2-2} & 231 \end{array}$$

We repeat this process until the remainder is zero.

$$\begin{array}{r|r} & 1 \\ \cline{2-2} 231 & 252 \\ & -\ \ 231 \\ \cline{2-2} & 21 \end{array} \quad \rightarrow \quad \begin{array}{r|r} & 11 \\ \cline{2-2} 21 & 231 \\ & -\ \ 21 \\ \cline{2-2} & 21 \\ & -\ \ 21 \\ \cline{2-2} & 0 \end{array}$$

The last divisor (21) is the gcd.

Space for Program and output ▼

Space for Program and output ▼

6. Write a program to process a collection of the speeds of vehicles. Your program should count and print the number of vehicles moving at a high speed (90 miles/hour or higher), the number of vehicles moving at a medium speed (50-89 miles/hour), and the number of vehicles moving at a slow speed (less than 50 miles/hour). It should also display the category of each vehicle. Test your program on the following data in a file:
`43 23 54 57 68 67 51 90 33 22 11 88 34 52 75 12 78 32 89 14 65 67 97 53 10 47 34`

    a. Also code to display the average speed of a category of vehicle (a real number) at the end of the run.

    b. Store the data into a file **`vspeed.txt`**. *Use input redirection to read all numbers from that file. (i.e.* **`./a.out`** *<* **`vspeed.txt`***)*

    c. While reading the input from the file, apply the idea of **`scanf`** function returns. *The* **`scanf`** *returns: (1) On success, this function returns the number of input items successfully matched and assigned (ii) The value* ***EOF*** *is returned if the end of input is reached before either the first successful conversion or a matching failure occurs.*

Space for Program and output ▼

Space for Program and output ▼

7. A baseball player’s batting average is calculated as the number of hits divided by the official number of at-bats. In calculating official at-bats, walks, sacrifices, and occasions when hit by the pitch are not counted. Write a program that takes an input file containing player numbers and batting records. Trips to the plate are coded in the batting record as follows: H-hit, O-out, W-walk, S-sacrifice, P-hit by pitch. The program should output for each player the input data followed by the batting average. Each batting record is followed by a newline character. **Your program should not use any kind of array and array processing**.

```
Sample input file:

12 HOOOWSHHOOHPWWHO
4 OSOHHHWWOHOHOOO
7 WPOHOOHWOHHOWOO
```

```
Corresponding output:

Player 12’s record: HOOOWSHHOOHPWWHO
Player 12’s batting average: 0.455
Player 4’s record: OSOHHHWWOHOHOOO
Player 4’s batting average: 0.417
Player 7’s record: WPOHOOHWOHHOWOO
Player 7’s batting average: 0.364
```

Space for Program and output ▼

Space for Program and output ▼

8. Write a program to process a collection of scores obtained by students of a class of certain strength. Your program should count and print the number of students with Grade A (80 and higher), Grade B(65-79), Grade C(40-64) and Grade F(39 and below). Ensure that the entered scores must remain in between 0 and 100(inclusive). Test your program on the following data:

```
8
23 67 65 12
89 32 17 45
41 58 60 78
82 88 19 22
70 88 41 89
78 79 72 68
74 59 75 81
44 59 23 12
```

   a. Read the same input from a file using input redirection. First line represents number of students and rest of the lines represent the marks obtained by each student in 4 subjects.

   b. Display average score and grade of each student in form of a table.(***Hint:*** *Average score of a student*=$(m_1 + m_2 + m_3 + m_4)/4$. *where* $m_i, i = 1, 2, 3, 4$ *represent mark in subject* i *and calculate grade according to the specified condition given above.*

Space for Program and output ▼

Space for Program and output ▼

9. Design a C program to display the following pattern;

```
A B C D E F G F E D C B A
A B C D E F   F E D C B A
A B C D E       E D C B A
A B C D           D C B A
A B C               C B A
A B                   B A
A                       A
```

Write/paste your code here ▼

10. The natural logarithm can be approximated by the following series.

$$\frac{x-1}{x}+\frac{1}{2}\left(\frac{x-1}{x}\right)^2+\frac{1}{2}\left(\frac{x-1}{x}\right)^3+\frac{1}{2}\left(\frac{x-1}{x}\right)^4+\ldots$$

If x is input through the keyboard, write a program to calculate the sum of first nine terms of this series.

Space for Program and output ▼

11. Write a menu driven program which has following options:

    1. Factorial of a number.
    2. Prime or not
    3. Odd or even
    4. Exit

    Use input-validation loop and program should terminate only when option 4 is selected.

Space for Program and output ▼

Space for Program and output ▼