Part A	Birla Institute of Technology & Science, Pilani First Semester 2022-23 CS F111 – Computer Programming Comprehensive Examination	Set X

==========

11/02/2023 **Max. Marks: 36M** **Duration: 60 mins**

==========

ID No: **Name:**

==========

Instructions:

- Write the answers only in the space provided.
- Don't let your answers flow outside the boxes.
- **The marking is strictly binary.**
- Over-written answers of any kind will not be accepted for rechecks.
- You must write only on C statement, wherever there is a blank followed by a semi colon.

Invigilator's Signature:

==========

Recheck requests (write in bullets for each question)	Remarks

1. What is a memory leak? Explain with an example. **3M**

2. The following program computes the sum of the coefficients of two polynomials using **sumPolynomial()** function. Each polynomial is a cubic polynomial. Example: **$10x^3+8x^2+25x+1$**. The structure given below is used to define such polynomials, wherein, the corresponding coefficients are stored in the fields: **cube_x**, **square_x**, **x** and **c**. The coefficients for both the polynomials are taken as input from the user. Fill the blanks with correct code segment to complete the functions. **8M**

```
#include <stdio.h>
#include<malloc.h>

typedef struct {
      int cube_x, square_x, x, c;
} polynomial;
```

```
polynomial sumPolynomial(polynomial a, polynomial b){
  // compute the sum of coefficient of polynomials

  _____;

  return a;
}

int main() {
  polynomial *ptr;
  ptr = (polynomial *) malloc(3 * sizeof(polynomial));
  // read co-efficients of the first two polynomials

  _____;

  _____;

  ptr[2]= sumPolynomial(ptr[0], ptr[1]);
  // print the coefficients of the new polynomial

  _____;

  free(ptr);
  return 0;
}
```

3. What will be output of the following function, if **fun (1, 5**) is called from the main function? **3M**

```
int fun (int a, int b)
{
  if (a==b) return (1);
  else if (a>b) return(0);
  else return (a+fun(a+1, b));
}
```

Ans:_____

4. Your program has to to sort a few numbers, which are to be taken as input using the command line arguments. For example, I would run my program using the following command: **$ ./a.out 66 22 44 33 88 55**. Complete the program given below, which creates an array to store the input numbers given using command line arguments and sorts that array. **4M**

```
int main(int argc, char *argv[]){
  int i, j, a; int n[argc];

  for (_____;_____;_____)
  {

    _____;
  }
  / *rest of the program sorts the above array. You can assume that this code
     is given to you.*/
}
```

5. What will be the output of the following program? **3M**

```
#include<string.h>
#include <stdio.h>
int main()
{
    char a[15]; char *s="Hello BITS";
    int len=strlen(s);
    for(int i=0;i<len; i++)
      a[i]=s[len-1-i];
    printf("%s",a);
    return 0;
}
```

Ans:_____;

6. Complete the following C function to remove extra white spaces from a given string.
   E.g, if the input string is **" My name is Hari "**, the output should be **"My name is Hari"**. **6M**

```
char *RemoveExtraSpaces(const char* Str)
{
    int i=0, j=0; //use these variables as loop iterators
    char *NewStr;
    NewStr = (char *)malloc(100*sizeof(char));

    while(Str[i] != '\0')
    {
```

```
        // write your answer in this box. Don't declare additional variables.
```

```
        NewStr[j] = Str[i];
        i++; j++;
    }
    //NULL terminate the new string
    NewStr[j] = '\0';

    return NewStr;
}
```

7. The following is an incomplete function in C to compare two files character by character. The function returns 0 if both files are equivalent, otherwise returns -1 and sets ***line*** (line number) and ***col*** (column number or the position of that mismatch character in that ***line***) where both files differ. **6M**

```
int compareFile(FILE * fPtr1, FILE * fPtr2, int * line, int * col)
{
    char ch1, ch2;

    *line = 1;
    *col  = 0;

    do
    {
        //Input character from both files
        ch1 = fgetc(fPtr1);
        ch2 = fgetc(fPtr2);

        if (ch1 == '\n')
        {
            _____________________________;

            _____________________________;
        }

        if (ch1 != ch2)

            _____________________________;

        *col  += 1;

    } while (ch1 != EOF && ch2 != EOF);

    if (ch1 == EOF && ch2 == EOF)
        return 0;
    else
        return -1;
}
```

8. Give three advantages of using dynamically allocated arrays vs using static arrays.

Advantage 1
Advantage 2
Advantage 3