Part B

Birla Institute of Technology & Science, Pilani
First Semester 2023-24
CS F111 – Computer Programming
Comprehensive Examination

===

13/12/2023 **Max. Marks: 55M** **Duration: 105 minutes**

===

ID No: **Name:**

===

Instructions:

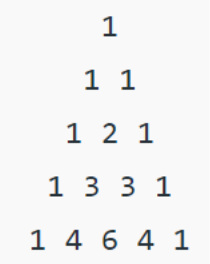
Invigilator's Signature:

- This is an in-built question paper. Write the answers only in the space provided.
- Don't let your answers flow outside the boxes.
- **The marking is strictly binary. Each blank will be awarded marks only if fully correct.**
- **Over-written answers of any kind will not be accepted for rechecks.**
- Assume that the necessary standard libraries, and the wrapper (main) functions exist wherever required.

===

Recheck requests (write in bullets for each question)	Remarks

1. Your task is to complete the function ***printPascalsTriangle*** to print Pascal's Triangle up to the specified number of rows ***n*** in the code below. Pascal's Triangle is a mathematical construct where each number (coefficient) is the sum of the two numbers directly above it in the previous row. The triangle begins with the number 1 at the top.



```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

[3+3+2 =8M]

```
void printPascalsTriangle(int n) {

  for (int i = 0; i < n; i++) {
    int coefficient = 1;

    for (________________________________) {  //Add leading spaces for alignment
      printf("  ");
    }

    for (____________________________) {  //Print the numbers (coefficients)
      printf("%d ", coefficient);

      ______________________________________________   //Update the coefficient for the next term
    }
    printf("\n");
  }
}
```

2. In a database of football players, a structure is defined with the name, jersey number, country, number of goals in NUM_MATCHES matches of a tournament and total goals of a player. Based on the total number of goals, the following incomplete program displays the details of the highest goal scorer. Fill in the blanks accordingly. **[7+7+7 = 21M]**

```
#define NUM_MATCHES 3
typedef struct {
   char name[20];
   int jersy_no;
   int goals[NUM_MATCHES];
   char country[20];
   int total_goals;
} Player;

void Score(Player p[], int n) {  //Function to calculate the total goals of each player
   for(int k=0;k<n;k++)
   {
       ________________________________________

      ________________________________________

      ________________________________________
   }
}

void readInfo(Player p[], int n) {  //Function to read each player's info
   for(int i=0;i<n;i++)
   {
      printf("Enter the name, jersy_no, and country name of each player\n");

      ________________________________________________________________
      printf("Enter the goal records for each player\n");

     _________________________________________________________________

     _________________________________________________________________
   }
   Score(p,n);
}

int maxGoals(Player p[], int n) {  //Function to return the index (pos) of the player with maximum goals
   int max=p[0].total_goals, pos=0;

   for(____________________________){

      _________________________________________

      _________________________________________
```

```
        ________________________________________

        ________________________________________

        ________________________________________

    }
    return pos;
}

int main() {
    int num_players;
    scanf("%d", &num_players);
    Player p[num_players];
    readInfo(p, num_players);
     int k=maxGoals(p, num_players);
    printf("Details of the highest Goal Scorer\n");
    printf("Name: %s \t jersy_no:%d \t Country:%s\t Total Goals:%d\n ", p[k].name, p[k].jersy_no, p[k].country,
p[k].total_goals);
    return 0;
}
```

3. Complete the following incomplete program to concatenate two strings using pointers. **[3+3 = 6M]**

```
#define MAX_SIZE 100 // Maximum string size
int main() {
    char str_A[MAX_SIZE], str_B[MAX_SIZE];
    char * str1 = str_A;
    char * str2 = str_B;

    /* Input two strings from user */
    printf("Enter first string: ");
    gets(str_A);
    printf("Enter second string: ");
    gets(str_B);

    while(_________________________);  //Move till the end of str_A

    while(_________________________); //Copy str_B to str_A
    printf("Concatenated string = %s", str_A);
    return 0;
}
```

4. The following C function ***printMiddle*** prints the middle element of a linked list using the concept of slow and fast pointers. The slow pointer moves one node at a time, while the fast pointer moves two nodes at a time. Complete the function accordingly. **[2*3 = 6M]**

```
struct Node {
    int data;
    struct Node* next;
};
```

```
// Function to get the middle of the linked list. For even number of nodes, return the first middle node
void printMiddle(struct Node *head) {
  struct Node *slow_ptr = head;
  struct Node *fast_ptr = head;

  if (head!=NULL) {
    while (_____________________________________________) {

      ______________________________________________

      ______________________________________________
    }
    printf("The middle element is %d \n", slow_ptr->data);
  }
}
```

5. Fill in the blanks for the program given below: **[2+2=4M]**

```
int *alloc_mem_and_set_value_as_4() {

  int *p;

  _________________________________________________; // Allocate memory dynamically for p

  __________________________________________; // Set the value of the location allocated to 4

  return p;

}
```

6. Total how many int elements are present in the following dynamic 2D array? _________________ **[2M]**

```
int **arr2d = (int **)malloc(3 * sizeof(int *));
for (int i = 0; i < 3; i++)
arr2d[i] = (int *)malloc((i/2 + i + 3) * sizeof(int));
```

7. For each of the following ((a) to (e)), choose the most appropriate declarations from the options given below in 1 to 12. **Just write the option from 1 to 12, no need to write the whole declaration.** **[5*2 = 10M]**
   a) An entity that can contain the address of a float variable. ____________________
   b) An entity which points to a float array of hundred values. ____________________
   c) An entity that can contain the addresses of hundred float variables. ____________________
   d) An entity that can store a scalar value at every unit cubic inch in a space whose dimensions are 10 inches long, 30 inches wide and 5 inches tall. ____________________
   e) An entity that can store the 3d positional coordinates at unit length in three dimensions along with a scalar value of a space which is 10 inches long, 30 inches wide and 5 inches tall. ____________________

```
typedef struct
{
  int length;
  int breadth;
  int height;
  float val;
} ELEM;
```

1. float *p;	2. float **p;	3. ELEM *p;	4. ELEM **p;
5. float p[100];	6. float *p[100];	7. float (*p)[100];	8. float p[10][30][5];
9. ELEM p[10][30][5];	10. ELEM p[100];	11. ELEM *p[100];	12. ELEM (*p)[100];

****