

Indian Institute of Information Technology, Sri City Chittoor

Computer Programming - Mid Term Examination 2

Duration: 1.5 Hrs

Date: 24-01-2023

Max marks: 30

Answer all questions
For MCQs write option and full answer
Part A

(10x1=10 marks)

For the questions 1-8, choose the correct output.

1.

```
#include<stdio.h>
int main() {
    int a = 3;
    switch(a) {
        L1: case 1: printf("IIIT SRICITY\n");
        case 2: printf("TADA\n"); break;
        case 3: printf("CSE\n");
        default:
            printf("IN DEFAULT\n");
            goto L1;
    }
    return 0;
}
```

☒ A. CSE
IN DEFAULT
IIIT SRICITY
TADA

☐ B. Infinite loop

☐ C. CSE

☐ D. IIIT SRICITY
TADA

2.

```
#include<stdio.h>
int main() {
    int i,n,a;
    for(i=0,n=scanf("%d",&a);i<5;n--,i++);
    printf("%d %d", i, n);
    return 0;
}
```

☒ A. 5 -5

☐ B. 5 n-a

☒ C. 5 -4

☐ D. Compilation error

0 1
1 0
2 -1
3 -2
4 -3
5 -4

3.

```
#include<stdio.h>
int main() {
    int count = 10, n=4;
    int arr[n];
    for(int i=0;i<n;i++) {
        for(int j=0;j<i;j++) {
            count++;
        }
        arr[i] = count++;
        printf("%d ", arr[i]);
    }
}
```

☐ A. 10 12 15 20

☐ B. 10 12 14 16

☐ C. 10 11 12 13

☒ D. None

10 12 15 19

4. `#include<stdio.h>`
`int main () {`
`int n=5, i=0;`
`int arr[n];`
`char c = 'u';`
`do {`
`arr[i++] = printf("%c ", --c);`
`} while(i < n);`

`for(int i=0; i<n; i++) {`
`printf("%d ", arr[i]);`
`}`

`return 0;`
`}`

A. pqrst22222 B. pqrst11111 C. tsrqp11111 D. tsrqp22222

5. `#include<stdio.h>`
`int main() {`
`int c=100, n=3;`
`int S[3][3] = {{1,2,3},{4,5,6},{7,8,9}};`
`for(int i=0 ; i<n ; i++) {`
`for(int j=0 ; j<n ; j++){`
`if (i != j) {`
`S[i][j] = S[i][j] + S[j][i];`
`S[j][i] = S[i][j] - S[j][i];`
`S[i][j] = S[i][j] - S[j][i];`
`} else {`
`S[i][j] = c;`
`c++;`
`}`
`}`
`}`
`return 0;`
`}`

123
456
789

103
256
789

0.1

A. 100 47
2 101 8
3 6 102

B. 100 23
4 101 6
7 8 102

C. 1 2 3
4 5 6
7 8 9

D. 100 1 2
3 101 4
5 6 102

6. `#include <stdio.h>`
`float get(){`
`return 1;`
`}`
`void main() {`
`int y = 0;`
`y = get();`
`printf("%d\n", y);`
`}`

- A. Compile time error
- B. 1
- C. 0
- D. Runtime error

```

#include<stdio.h>
int reverse(int no) {
    if(no == -1) return 0;
    else printf("%d ", no);
    reverse(--no);
}
int main() {
    reverse(5);
    return 0;
}

```

- A. 5 4 3 2 1
- B. 1 2 3 4 5
- ☒ C. 5 4 3 2 1 0
- D. Infinite loop

8.

```

#include <stdio.h>
#define cube(a) a*a*a

int main() {
    int x;
    x = 27/cube(3);
    printf("%d\n", x);
    return 0;
}

```

- ☒ A. 1
- B. 27
- C. 81
- D. 9

9. What does the base-case do in a recursive function?

- ☒ A. Terminates the recursion
- B. Facilitates the recursion
- C. Accumulates the results
- D. Raises an exception

10. Let f be a function defined as given below. What is the value returned by the call $f(237)$?

```

int f(int n) {
    if (n < 5) return n;
    return f(n/3);
}

```

- A. 0
- B. 3
- ☒ C. 2
- D. 118

Handwritten calculation for $f(237)$:

$$\begin{array}{r}
 3 \overline{) 237} \\
 \underline{3 \times 79} \\
 26 \\
 \underline{3 \times 26} \\
 8 \\
 \underline{3 \times 8} \\
 2
 \end{array}$$

(Questions continue in next page)

Part B

Q1. Write the program for the following pattern.

Input: n=5

Output:

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Input: n = 3

Output :

```
1 2 3
1 2
1
```

(10 marks)

Q2. Consider an array of n floating point numbers where n can be randomly chosen in $[30, 50]$.

The values of this array are randomly generated floating point numbers in the range $[1.0, 9.0]$, each with 6 digits of precision. Now write the following functions to perform the specified tasks:

- Write a recursive function that considers two subsequent floating point numbers whose difference is less than or equal to m where m can be specified by the user.
- Write a recursive function to identify and print the floating point numbers that contain the integer part of the number in its fractional part. For example, let us consider any two elements of the array, say 4.543421 and 4.523123. This function should print the first one 4.543421 as it contains the integer part, the number 4, in its fractional part as well (at least one occurrence) and in case of the second one, the fractional part of 4.523123 does not contain 4.
- Write a recursive function that converts the given array into two arrays in such a way that the first one contains all floating point numbers $< k$ and the second array contains all floating point numbers $\geq k$. Here k can be specified by the user.

(10 marks)
