**General Instructions:**

* **Follow the instructions given in each section.**
* **Make sure that you attempt the questions in order.**

**SECTION-A (10\*1 mark=10 marks)**

***(All questions are compulsory)***

Q1 Which of the following is true about tail recursion?

a) It is faster than normal recursion

b) It is slower than normal recursion

c) It takes less memory than normal recursion

d) It takes more memory than normal recursion

Answer: c) It takes less memory than normal recursion

Q2 1. Which of the following is the correct way to write a tail recursive function?

a)return\_type function\_name(arguments){

if(base\_case)

return some\_value;

function\_name(arguments);

}

b) return\_type function\_name(arguments, int acc){

if(base\_case)

return acc;

return function\_name(arguments, acc);

}

c) return\_type function\_name(arguments, int acc){

if(base\_case)

return acc;

function\_name(arguments, acc);

}

d) None of the above

Answer: b) return\_type function\_name(arguments, int acc){

if(base\_case)

return acc;

return function\_name(arguments, acc);

}

Q3 What is the maximum number of recursive calls that can be made in a program?

1. It depends on the size of the stack
2. It depends on the size of the heap
3. It is unlimited
4. None of the above

Answer: a) It depends on the size of the stack

Q4 1. Which of the following is an example of a nested struct in C?

A) struct point { int x; int y; };

B) struct student { char name[20]; int age; };

C) struct rectangle { struct point top\_left; struct point bottom\_right; };

D) None of the above

Answer: C) struct rectangle { struct point top\_left; struct point bottom\_right; };

Q5 1. Which of the following is a valid way to initialize a struct in C?

A) struct s = {1, 2, 3};

B) struct s = { .x=1, .y=2 };

C) struct s = {1, .y=2};

D) All of the above

Answer: D) All of the above

Q6 1. What is the keyword used to allocate memory dynamically for a struct in C?

A) malloc

B) free

C) realloc

D) none of the above

Answer: A) malloc

Q7 1. What is the default value of the first element of an enumeration in C?

a. 0

b. 1

c. -1

d. None of the above

Answer: a

Q8 1. 1. Can two elements of an enumeration have the same value in C?

a. Yes

b. No

c. Case specific

d. Enum is not a datatype

Answer: a

Q9 1. What function is used to write a string to a file in C?

a) fprintf()

b) fputc()

c) fputs()

d) fread()

Answer: c) fputs()

Q10 1. What function is used to read a line of text from a file in C?

a) fread()

b) fgetc()

c) fgets()

d) fprintf()

Answer: c) fgets()

**SECTION-B (5\*2 mark=10 marks)**

***(All questions are compulsory)***

Q11

What is the output of the following recursive function when called with an argument of 4?

int sum(int n) {

if (n == 0) {

return 0;

}

return n + sum(n-1);

}

A. 10

B. 16

C. 20

D. 24

Q12 What is the output of the following code?

#include <stdio.h>

struct person {

char name[20];

int age;

float salary;

};

int main() {

struct person p1 = { "John", 25, 2500.50 };

struct person p2 = { "Mary", 30, 3000.00 };

p1 = p2;

printf("%s %d %.2f\n", p1.name, p1.age, p1.salary);

return 0;

}

A. Mary 30 3000.00

B. John 25 2500.50

C. Mary 25 2500.50

D. Error

Correct answer: A

Q13 Which of the following statements declares a typedef for a structure called student?

a. typedef student { int id; char name[50]; }

b. struct student { int id; char name[50]; }

c. typedef struct { int id; char name[50]; } student

d. typedef { int id; char name[50]; } student

Correct answer: c

Q14 What is the output of the following code snippet?

#include <stdio.h>

enum months { JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC };

int main() {

enum months m1 = APR, m2 = NOV;

if(m1 == m2) printf("Equal");

else printf("Not equal");

return 0;

}

a) Equal

b) Not equal

c) Compiler error

d) Runtime error

Answer: b

Q15 What is the output of the following program?

#include <stdio.h>

union data {

int i;

float f;

char c;

};

int main() {

union data d;

d.i = 65;

printf("d.i = %d\n", d.i);

d.f = 3.14;

printf("d.f = %f\n", d.f);

d.c = 'A';

printf("d.c = %c\n", d.c);

return 0;

}

a) d.i = 65, d.f = 3.14, d.c = A

b) d.i = garbage value, d.f = 3.14, d.c = A

c) d.i = 65, d.f = garbage value, d.c = A

d) Compiler error

Correct answer: a

**SECTION-C(Coding Question) (2x5 marks=5 marks)**

Q16. Create a structure named "student" that includes name, age, and grade. Write a program to create an array of five "student" structures, input data for each student, and display the details of the student with the highest grade.

**Input:**

One line with integer **n** giving the size of the array.

The next n line contains a string name, integer age and integer grade.

**Constraints:**

1<=**strlen(name)**<=1000

1<=**agei** <=90

0.00<= grade <= 10.00

**Output:**

**display name, age and grade of topper space separated in a single line.**

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 3  19 8.6 devansh  20 7.9 ashu  21 8.5 ayush  devansh 19 8.60  Explanation: for 0th student 19,8.6,devansh is age, grade, name respectively.  for 1st student 20,7.9,is age, grade, name respectively.  for 2nd student 21,8.5,ayush is age, grade, name respectively.  among devansh, ashu and ayush devansh got the highest score. | devansh 19 8.60 |
| STC2 | 2  20 9.3 rahul  21 7.6 arzoo | rahul 20 9.30 |

**Solution 16:**

#include<stdio.h>

#include<string.h>

struct student {

char name[20];

int age;

float grade;

};

int main() {

int n;

scanf("%d",&n);

struct student s[n];

int i, max = 0;

for(i=0; i<n; i++) {

scanf("%s %d %f", s[i].name, &s[i].age, &s[i].grade);

if(s[i].grade > s[max].grade) {

max = i;

}

}

printf("%s %d %0.2f", s[max].name, s[max].age, s[max].grade);

return 0;

}

Test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| TC1 | 1  20 7.2 devansh | devansh 20 7.20 |
| TC2 | 2  7 8.3 pinki  8 9.1 ashu | ashu 8 9.10 |
| TC3 | 3  10 9.0 ayush  20 7.2 devanshi  7 8.9 ayushi | ayush 10 9.00 |
| TC4 | 4  7 8.3 pinki  8 9.1 ashu  10 9.0 ayush  20 7.2 devanshi | ashu 8 9.10 |
| TC5 | 5  8 9.1 ashu  10 9.0 ayush  20 7.2 devanshi  7 8.3 pinki  8 9.1 ashu | ashu 8 9.10 |

Q17. Write a C program to find the factorial of a number using recursion.

**Input:**

One line with integer **n**.

**Constraints:**

0<= **n** <=25

**Output:**

**display factorial of n**

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 3 | 6 |
| STC2 | 1 | 1 |

**Solution 17:**

#include<stdio.h>

#include<string.h>

struct student {

char name[20];

int age;

float grade;

};

int main() {

int n;

scanf("%d",&n);

struct student s[n];

int i, max = 0;

for(i=0; i<n; i++) {

scanf("%s %d %f", s[i].name, &s[i].age, &s[i].grade);

if(s[i].grade > s[max].grade) {

max = i;

}

}

printf("%s %d %0.2f", s[max].name, s[max].age, s[max].grade);

return 0;

}

Test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| TC1 | 10 | 3628800 |
| TC2 | 12 | 479001600 |
| TC3 | 13 | 1932053504 |
| TC4 | 14 | 1278945280 |
| TC5 | 15 | 2004310016 |

**SECTION-D (Coding Question)(1x10 mark=10 mark)**

**Q18 Write a program to implement Tower Of Hanoi using Recursion.**

**Sample Input:Solve Tower of Hanoi problem for 3 discs**

**Sample Output: Move disc 1 from rod A to rod C Move disc 2 from rod A to rod B Move disc 1 from rod C to rod B Move disc 3 from rod A to rod C Move disc 1 from rod B to rod A Move disc 2 from rod B to rod C Move disc 1 from rod A to rod C**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | **Solve Tower of Hanoi problem for 3 discs** | **Solve Tower of Hanoi problem for 4 discs** | **Solve Tower of Hanoi problem for 5 discs** |
| **Output** | **ove disc 1 from rod A to rod CMove disc 2 from rod A to rod BMove disc 1 from rod C to rod BMove disc 3 from rod A to rod CMove disc 1 from rod B to rod AMove disc 2 from rod B to rod CMove disc 1 from rod A to rod C** | **Move disc 1 from rod A to rod BMove disc 2 from rod A to rod CMove disc 1 from rod B to rod CMove disc 3 from rod A to rod BMove disc 1 from rod C to rod AMove disc 2 from rod C to rod BMove disc 1 from rod A to rod BMove disc 4 from rod A to rod CMove disc 1 from rod B to rod CMove disc 2 from rod B to rod AMove disc 1 from rod C to rod AMove disc 3 from rod B to rod CMove disc 1 from rod A to rod BMove disc 2 from rod A to rod CMove disc 1 from rod B to rod C** | **Move disc 1 from rod A to rod C**  **Move disc 2 from rod A to rod B**  **Move disc 1 from rod C to rod B**  **Move disc 3 from rod A to rod C**  **Move disc 1 from rod B to rod A**  **Move disc 2 from rod B to rod C**  **Move disc 1 from rod A to rod C**  **Move disc 4 from rod A to rod B**  **Move disc 1 from rod C to rod B**  **Move disc 2 from rod C to rod A**  **Move disc 1 from rod B to rod A**  **Move disc 3 from rod C to rod B**  **Move disc 1 from rod A to rod C**  **Move disc 2 from rod A to rod B**  **Move disc 1 from rod C to rod B**  **Move disc 5 from rod A to rod C**  **Move disc 1 from rod B to rod A**  **Move disc 2 from rod B to rod C**  **Move disc 1 from rod A to rod C**  **Move disc 3 from rod B to rod A**  **Move disc 1 from rod C to rod B**  **Move disc 2 from rod C to rod A**  **Move disc 1 from rod B to rod A**  **Move disc 4 from rod B to rod C**  **Move disc 1 from rod A to rod C**  **Move disc 2 from rod A to rod B**  **Move disc 1 from rod C to rod B**  **Move disc 3 from rod A to rod C**  **Move disc 1 from rod B to rod A**  **Move disc 2 from rod B to rod C**  **Move disc 1 from rod A to rod C** |

**Solution:**

**#include <stdio.h>**

**void tower\_of\_hanoi(int num\_discs, char from\_rod, char to\_rod, char aux\_rod);**

**int main() {**

**// Test Case 1: Solve Tower of Hanoi problem for 3 discs**

**printf("Test Case 1: Solve Tower of Hanoi problem for 3 discs\n");**

**tower\_of\_hanoi(3, 'A', 'C', 'B');**

**// Test Case 2: Solve Tower of Hanoi problem for 4 discs**

**printf("\nTest Case 2: Solve Tower of Hanoi problem for 4 discs\n");**

**tower\_of\_hanoi(4, 'A', 'C', 'B');**

**// Test Case 3: Solve Tower of Hanoi problem for 5 discs**

**printf("\nTest Case 3: Solve Tower of Hanoi problem for 5 discs\n");**

**tower\_of\_hanoi(5, 'A', 'C', 'B');**

**return 0;**

**}**

**void tower\_of\_hanoi(int num\_discs, char from\_rod, char to\_rod, char aux\_rod) {**

**if (num\_discs == 1) {**

**printf("Move disc 1 from rod %c to rod %c\n", from\_rod, to\_rod);**

**return;**

**}**

**tower\_of\_hanoi(num\_discs-1, from\_rod, aux\_rod, to\_rod);**

**printf("Move disc %d from rod %c to rod %c\n", num\_discs, from\_rod, to\_rod);**

**tower\_of\_hanoi(num\_discs-1, aux\_rod, to\_rod, from\_rod);**

**}**