**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 What is the purpose of a typedef in C?

a. To create a new data type with a different name

b. To define a new variable

c. To declare a new function

d. None of the above

Answer: a

Q2 What is the syntax for declaring a typedef for a union in C?

a. typedef union { } union\_name;

b. typedef struct { } struct\_name;

c. typedef enum { } enum\_name;

d. None of the above

Answer: a

Q3 What is the syntax for accessing a member of a union in C?

a. union\_name.member\_name

b. union\_name->member\_name

c. union\_name[member\_name]

d. None of the above

Answer: a

Q4 1. Which of the following statements is true about unions in C?

a. Only one member of a union can be used at a time

b. All members of a union can be used at the same time

c. Union members can be of different sizes

d. Union members must be of the same size

Answer: a

Q5 1. What is the default mode when opening a file in C?

a) read mode

b) write mode

c) append mode

d) binary mode

Answer: a) read mode

Q6 1. What function is used to open a file in C?

a) read()

b) write()

c) fopen()

d) fclose()

Answer: c) fopen()

Q7 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()

Q8 1. Which of the following is true about recursion?

a) It is faster than iteration

b) It is slower than iteration

c) It takes more memory than iteration

d) It takes less memory than iteration

Answer: c) It takes more memory than iteration

Q9 1. 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

Q10 What is the maximum number of recursive calls that can be made in a program?

It depends on the size of the stack

It depends on the size of the heap

It is unlimited

None of the above

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

**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 5?

int baz(int n) {

if (n == 0) {

return 1;

}

return 2 \* baz(n-1);

}

A. 1

B. 2

C. 16

D. 32

Q12 What is the output of the following code?

#include <stdio.h>

struct person {

char \*name;

int age;

};

int main() {

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

struct person p2 = p1;

p2.age = 30;

printf("%d\n", p1.age);

return 0;

}

A. 25

B. 30

C. Error

D. None of the above

Correct answer: A

Q13 What is the difference between a struct and a typedef struct declaration?

a. There is no difference

b. A typedef struct declaration creates a new type name for the structure

c. A struct declaration creates a new type name for the structure

d. A typedef struct declaration allows access to private members of the structure

Correct answer: b

Q14 What is the output of the following code snippet?

#include <stdio.h>

enum { A = 10, B, C = 5, D };

int main() {

printf("%d %d %d %d", A, B, C, D);

return 0;

}

a) 10 11 5 6

b) 10 11 5 5

c) 10 10 5 6

d) 10 10 5 5

Answer: a

Q15 What is the output of the following program?.

#include <stdio.h>

#include <string.h>

struct book {

char title[50];

int pages;

float price;

};

int main() {

struct book my\_book = {"A Tale of Two Cities", 307, 12.50};

printf("%d\n", strlen(my\_book.title));

return 0;

}

a) 21

b) 22

c) 20

d) None of the above

Correct answer: c

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

Q16. Define a structure named Time to represent the time of day (hour, minute, and second). Write a program to read two times from the user, subtract them, and display the result on the console. see test cases for more clarification.

**Input:**

two lines with three integers hour, minute, second.

**Constraints:**

0<=**minute,second**<=60

0<=**hours**<=24

**Output:**

**print the resultant time in the format hours:minutes:seconds**

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 12 23 45  21 43 45 | 9:20:0 |
| STC2 | 20 10 10  2 20 30 | 18:10:20 |

**Solution 16:**

#include <stdio.h>

struct Time {

int hour;

int minute;

int second;

};

int main() {

struct Time t1, t2, result;

scanf("%d%d%d", &t1.hour, &t1.minute, &t1.second);

scanf("%d%d%d", &t2.hour, &t2.minute, &t2.second);

result.second = t1.second + t2.second;

result.minute = t1.minute + t2.minute + result.second/60;

result.hour = t1.hour + t2.hour + result.minute/60;

result.second %= 60;

result.minute %= 60;

printf("%d:%d:%d\n", result.hour, result.minute, result.second);

return 0;

}

Test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| TC1 | 12 40 60  21 23 43 | 9:17:17 |
| TC2 | 0 0 0  0 0 0 | 0:0:0 |
| TC3 | 12 23 34  0 0 0 | 12:23:34 |
| TC4 | 23 34 45  12 23 45 | 11:11:0 |
| TC5 | 20 30 40  20 30 40 | 0:0:0 |

Q17. Imagine you're a mathematician studying the properties of numbers. You come across a unique number and its square root, such that when the number is raised to the power of its square root, a fascinating result occurs. You have given a number and its square root. You have to determine what the result of this calculation would be?

As answers can be very large, print the result modulo 109 + 7.

input:

two space separated integer values a, sqrt(a)

**constraints**:

1 <= N,sqrt(N) <= 10^7

Note: sqrt(N) will for sure be an integer value.

Output:

print a single integer result

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 10 1  Explanation: The square root of 1 is 1  and after raising power of 10 by 1  we get 10 which gives remainder as  10 by dividing 1000000007. | 10 |
| STC2 | 4 2 | 16 |

**Solution 17:**

#include <stdio.h>

# define mod 1000000007

long long solve(int N,int R)

{

if(R==0) return 1;

long long temp = solve(N,R/2);

temp=(temp\*temp)%mod;

if(R%2==1)

{

return (temp\*N)%mod;

}

return temp;

}

int main()

{

long long N,R;

scanf("%ld%ld",&N,&R);

long long ans =solve(N,R);

printf("%ld",ans);

}

Test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| TC1 | 16 4 | 65536 |
| TC2 | 64 4 | 974740338 |
| TC3 | 100 10 | 4900 |
| TC4 | 1000000 1000 | 460300837 |
| TC5 | 210681 459 | 870165434 |

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

Q18 I**magine you are an astronaut preparing for a space mission. As part of the preparation, you need to know your weight on different planets to plan and manage your nutrition and exercise regimen during the mission. You use this program to input your current weight on Earth and obtain your weight on Mercury, Venus, Earth, and Mars planets.**

**Sample Input**:

Enter Your Weight : 50

**Sample Output**:

Your Weight on MERCURY is 18.925160 Your Weight on VENUS is 45.478370 Your Weight on EARTH is 50.000000 Your Weight on MARS is 19.215636

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 |
| Input | Enter Your Weight : 75 | Enter Your Weight : 100 | Enter Your Weight : 50 |
| Output | Your Weight on MERCURY is 28.387741 Your Weight on VENUS is 68.236554 Your Weight on EARTH is 75.000000 Your Weight on MARS is 28.583292 | Your Weight on MERCURY is 37.850321 Your Weight on VENUS is 90.956739 Your Weight on EARTH is 100.000000 Your Weight on MARS is 38.431273 | Your Weight on MERCURY is 18.925160 Your Weight on VENUS is 45.478370 Your Weight on EARTH is 50.000000 Your Weight on MARS is 19.215636 |

**Solution:**

**#include <stdio.h>**

**#include <math.h>**

**// Define a struct for a planet**

**typedef struct {**

**double mass; // in kg**

**double radius; // in meters**

**} Planet;**

**// Define constants for each planet**

**const Planet MERCURY = {3.303e+23, 2.324397e8};**

**const Planet VENUS = {4.869e+24, 6.0518e6};**

**const Planet EARTH = {5.976e+24, 6.37814e6};**

**const Planet MARS = {6.421e+23, 3.3972e6};**

**// Define functions for calculating surface gravity and weight on each planet**

**double surfaceGravity(Planet p) {**

**const double G = 6.67300E-11;**

**return G \* p.mass / (p.radius \* p.radius);**

**}**

**double surfaceWeight(Planet p, double otherMass) {**

**return otherMass \* surfaceGravity(p);**

**}**

**int main() {**

**// Prompt the user for their weight on Earth**

**printf("Enter Your Weight : ");**

**double earthWeight;**

**scanf("%lf", &earthWeight);**

**// Calculate the mass of the person based on their weight on Earth**

**double mass = earthWeight / surfaceGravity(EARTH);**

**// Print the person's weight on each planet**

**printf("Your Weight on MERCURY is %f\n", surfaceWeight(MERCURY, mass));**

**printf("Your Weight on VENUS is %f\n", surfaceWeight(VENUS, mass));**

**printf("Your Weight on EARTH is %f\n", surfaceWeight(EARTH, mass));**

**printf("Your Weight on MARS is %f\n", surfaceWeight(MARS, mass));**

**return 0;**

**}**