**10 MCQ (1 mark each)**

Q.1. A variable which is visible only in the function in which it is defined, is called \_\_\_\_\_\_\_\_\_\_.

A. static variable

B. auto variable

C. external variable

D. local variable

Q.2. What will be the result of the following code snippet?

#include <stdio.h>

void solve() {

char ch[10] = "abcdefghij";

int ans = 0;

for(int i = 0; i < 10; i++) {

ans += (ch[i] - 'a');

}

printf("%d", ans);

}

int main() {

solve();

return 0;

}

A) 45

B) 36

C) 20

D) 100

Q.3. The name of all functions end with a \_\_\_\_\_\_\_\_\_\_.

A) pair of parenthesis.

B) semicolon

C) braces

D) colon

Q.4. Output of the program given below is?

int i;

main()

{

printf("%d", i);

}

A. 1

B. 0

C. Garbage value

D. Null

Q.5. Is it necessary that in a function which accepts variable argument list there should be at least be one fixed argument?

A) Yes

B) No

Q.6. A self contained block of statements that perform a coherent task of some kind is called \_\_\_\_\_\_\_\_\_\_.

A) a Monitor

B) a Function

C) a Program

D) a Structure

Q.7. Which of the following statements are correct about the function?

int fun(int x)

{

int r = 1;

if(x==1)

return 1;

else

r = x \* fun(x-1);

return r;

}

A) The function calculates the 2 raise to power of x.

B) The function calculates the square root of x.

C) The function return the cube of x.

D) The function calculates factorial of x.

Q.8.In C language all function except main() can be recursive?

A) Yes

B) No

Q.9. What will be the output of the following code snippet?

#include <stdio.h>

void swap(int \*a, int \*b) {

int t = \*a;

\*a = \*b;

\*b = t;

}

void solve() {

int a = 3, b = 5;

swap(&a, &b);

printf("%d %d", a, b);

}

int main() {

solve();

return 0;

}

A) 3 5

B) 5 3

C) 5 5

D) 3 3

Q.10. Can we write a function that takes a variable argument list and passes the list to another function?

A) Yes

B) No

**5 MCQ (2 marks each)**

Q.1. The following program \_\_\_\_\_\_\_\_\_\_.

main()

{

inc(); inc(); inc();

}

inc()

{

static int x;

printf("%d", ++x);

}

a) prints 012

b) prints 123

c) prints 3 consecutive, but unpredictable numbers

d) prints 111

Q.2. What is the output of the following program?

main()

{

int a = 4;

change(a);

printf("%d", a);

}

change(a)

{

printf("%d", ++a);

}

a) 55

b)45

c) 54

d) 44

Q.3. What will be the output of the following code snippet?

#include <stdio.h>

void solve() {

int a = 3;

int res = a++ + ++a + a++ + ++a;

printf("%d", res);

}

int main() {

solve();

return 0;

}

A) 12

B) 24

C) 20

D) 18

Q.4. What will be the output of given code

main()

{

char ch[20];

ch = “Hi Hello";

printf("%s", ch);

}

A. error

B. Hi Hello

C. Hi

D. Hello

Q.5. The following program reports an error on compilation.

#include<stdio.h>

int main()

{

float i=10, \*j;

void \*k;

k=&i;

j=k;

printf("%f\n", \*j);

return 0;

}

A) True

B) False

**2 Coding Questions (5 mark each)**

Q.1. Player’s score

Developers at the mobile company 'TalkFree' have designed a game for the launch of their new product. In the game, the player is given a number. The player has to find out the difference between the number and the reverse of the number. The difference between the two numbers is the player's score. The number given to the player and the player's score can either be a negative or positive number. Write an function to find the player score.

**Sample Input 1**

64

**Sample Output 1**

18

**Sample Input 2**

5678

**Sample Output 2**

-3087

**Input Explanation**

The input consists of an integer, representing the number given to the player.

**Output Explanation**

Output is an single integer representing the player's score

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 10000000 | 9999 | 15698 | 2313 | 121 |
| **Output** | 9999999 | 0 | -73953 | -819 | 0 |

**#Solution**

#include<stdio.h>

long int reverse(long int n)

{

long int rev=0,rem,temp;

temp=n;

while(n)

{

rem=n%10;

rev=rev\*10+rem;

n/=10;

}

return temp-rev;

}

int main()

{

long int n;

scanf("%ld",&n);

printf("%ld",reverse(n));

return 0;

}

Q.2. Number of Distribution

The warehouse of an e-commerce company has limited stock available for each item due to a promotional sale. The company needs to distribute this stock to its distribution center in a particular manner. If the value of the available stock for an item is an even number, then the total stock is divided by 2. If it is an odd number, 1 is subtracted from it. The company wishes to know how many distributions can take place before the stock reaches 0.

Write an function to calculate the number of distributions that can take place before the stock reaches 0.

Example

Input

12

Output

5

Explanation

12 is even, 12/2 is 6

6 is even, 6/2 is 3

3 is odd, 3-1 is 2

2 is even, 2/2 is 1

1 is odd, 1-1 is 0

The total number of distributions that take place is 5. Hence the output is 5.

**Sample Input 1**

20

**Sample Output 1**

9

**Sample Input 2**

54

**Sample Output 2**

9

**Input Explanation**

The input consists of an integer, representing the stock available for the item (N).

**Output Explanation**

Print an integer value representing the number of distributions that can take place before the stock reaches 0.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 34 | 143 | 234 | 543 | 20 |
| **Output** | 7 | 12 | 12 | 15 | 6 |

**#Solution**

#include<stdio.h>

int fun(int n)

{

int count = 0;

while(n != 0)

{

if(n % 2 == 0)

{

n = n / 2;

count++;

}

else

{

n = n - 1;

count++;

}

}

return count;

}

int main()

{

int n, count = 0;

scanf("%d", &n);

printf("%d", fun(n));

return 0;

}

**1 Coding Question (10 mark)**

Q.1. Encryption key

The IT giant "SoftCompInfo" has decided to transfer its message through the network using a new encryption technique. The company has decided to encrypt the data using the non-prime number concept. The message is in the form of a number and the sum of non-prime digits present in the message is used as the encryption key. Write a function to determine the encryption key.

Example

Input

45673

Output

10

Explanation

The non-prime digits are 4 and 6. Hence the output is 4+6 = 10.

**Sample Input 1**

33512

**Sample Output 1**

0

**Sample Input 2**

468

**Sample Output 2**

18

**Input Explanation**

The input consists of a single integer value

**Output Explanation**

Output consists single integer value

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 6461 | 1001 | 33512 | 23456 | 48412 |
| **Output** | 16 | 0 | 0 | 10 | 12 |

**#Solution**

#include <stdio.h>

int nonprime(int);

int nonprime(int n)

{

int digit,i,sum=0,m;

while(n!=0)

{

digit=n%10;

n=n/10;

for(i=2;i<digit;i++)

{

if(digit%i==0)

{

sum=sum+digit;

break;

}

}

}

return sum;

}

int main()

{

int n;

scanf("%d",&n);

printf("%d",nonprime(n));

return 0;

}