**10 MCQ (1 mark each)**

Q.1. What error would the following function give on compilation?

fun(int a, int b)

{

int a = 20;

return a;

}

a) no error

b) The function should be defined as int fun(int a,int b)

c) Re-declaration of variable a

d) none of the above

Q.2. A function can be defined inside another function?

a) True

b) False

Q.3. max is a function that returns the larger of the two integers, given as arguments. Which of the following statements not finds the largest of three given numbers?

a) max(max(a, b), max(a, c))

b) max(a, max(a, c))

c) max(max(a, b), max(b, c))

d) max(b, max(a, c))

Q.4. What will be the value of x in the following code snippet?

#include <stdio.h>

void solve() {

int x = printf("Hello");

printf(" %d", x);

}

int main() {

solve();

return 0;

}

A) 10

B) 5

C) 1

D) 0

Q.5. Point out the compile time error in the program given below.

#include<stdio.h>

int main()

{

int \*x;

\*x=100;

return 0;

}

A) Error: invalid assignment for x

B) Error: suspicious pointer conversion

C) No error

D) None of above

Q.6. For the following statements will arr[3] and ptr[3] fetch the same character?

char arr[] = "India";

char \*ptr = "India";

A) Yes

B) No

Q.7. For a function receives variable number of arguments it is necessary that the function should receive at least one fixed argument.

A) True

B) False

Q.8. What will be the output of following code?

#include<stdio.h>

main()

{

float a = 10.2;

int \*p = a;

printf("%d",\*p);

}

A) 10

B) 10.2

C) error: incompatible type

D) no output

Q.9. Which of the statement is correct about the code?

int sum(int, int);

int (\*s)(int, int);

s = sum;

A) s is a pointer to a function sum which return integer

B) s is a function which return integer pointer

C) s is a function similar to sum function

D) none of the above

Q. 10. It is necessary that for the string functions to work safely the strings must be terminated with '\0'.

A) True

B) False

**5 MCQ (2 mark each)**

Q.1. What will be the output of the following code snippet?

#include <stdio.h>

void solve()

{

int a[] = {1, 2, 3, 4, 5};

int sum = 0;

for(int i = 0; i < 5; i++) {

if(i % 2 == 0) {

sum += \*(a + i);

}

else {

sum -= \*(a + i);

}

}

printf("%d", sum);

}

int main()

{

solve();

return 0;

}

A) 2

B) 15

C) sytax error

D) 3

Q.2. Which of the following statements are correct about the below declarations?

char \*p = "Sanjay";

char a[] = "Sanjay";

1: There is no difference in the declarations and both serve the same purpose.

2: p is a non-const pointer pointing to a non-const string, whereas a is a const pointer pointing to a non-const pointer.

3: The pointer p can be modified to point to another string, whereas the individual characters within array a can be changed.

4: In both cases the '\0' will be added at the end of the string "Sanjay".

A) 1, 2

B) 2, 3, 4

C) 3, 4

D) 2, 3

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

#include <stdio.h>

int fun(int \*a,int \*b)

{

\*a = \*a+\*b;

\*b = \*a-\*b;

\*a = \*a-\*b;

}

main()

{

int x = 10,y = 20;

fun(&x,&y);

printf("x= %d y = %d\n", x, y);

}

A) x=10 y=20

B) x=20 y=10

C) 20 10

D) error

Q.4. What will be the output of following code assuming that array begins at location 1002?

#include<stdio.h>

main()

{

int a[5] = {1, 2, 3, 4, 5};

int \*p = a;

printf("%d\t%d\t%d\t%d\t",\*p,0[a],a,p);

}

A) error

B) 1 1 1002 1002

C) 1 0 1002 0

D) 1 1 0 1002

Q.5. What will be the output of following code?

#include <stdio.h>

void main()

{

int const \*p = 5;

printf("%d", ++(\*p));

}

A) error

B) 5

C) 6

D) print address of p

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

Q.1. Prime Numbers in range

A school students want to check prime numbers within given range, write a c program to help them for the same. Where x will be starting range and y will be ending range, if x is greater than y or x is equal to y print 0, if x is smaller than y prints the prime numbers within mentioned range.

A number is said to be prime if it is divisible by 1 and the number itself.

**Sample Input 1**

50 90

**Sample Output 1**

53 59 61 67 71 73 79 83 89

**Sample Input 2**

60 6

**Sample Output 2**

0

**Input Explanation**

Input consists of two space separated integer value

First input will be taken as the value for x that is starting value

Second input will be taken as the value for y that is ending value

**Output Explanation**

Output can consists multiple space seperated integer value based upon input

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 100 150 | 45 7 | 15 60 | 55 3 | 25 25 |
| **Output** | 101 103 107 109 113 127 131 137 139 149 | 0 | 17 19 23 29 31 37 41 43 47 53 59 | 0 | 0 |

**#Solution**

#include <stdio.h>

int main()

{

int x, y, i, flag;

scanf("%d %d",&x, &y);

if (x > y || x==y)

{

printf("0");

}

else

{

while (x < y)

{

flag = 0;

for (i = 2; i <= x / 2; ++i)

{

if (x % i == 0)

{

flag = 1;

break;

}

}

if (flag == 0)

printf("%d ", x);

++x;

}

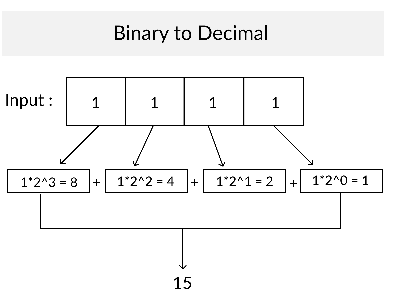
}

return 0;

}

Q.2. Binary to Decimal conversion

Ajit is working in IT company and he is getting binary values from computers, as a human he is not able understand binary values, so help Ajit to understand binary values by converting it into decimal values.



Consider the binary number from the last.

For the above mentioned example,

1 \* 2^0 = 1

1 \* 2^1 = 2

1 \* 2^2 = 4

1 \* 2^3 = 6

Decimal number = 1 + 2 + 4 + 6 = 15,

1111 in binary form is represented as 15 in decimal.

**Sample Input 1**

101011

**Sample Output 1**

43

**Sample Input 2**

110110

**Sample Output 2**

54

**Input Explanation**

Input consists of single integer value

**Output Explanation**

Output consists of single integer value

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 111111111 | 100011 | 1110011 | 1011001 | 100100100 |
| **Output** | 511 | 35 | 115 | 89 | 292 |

**#Solution**

#include <stdio.h>

#include<math.h>

int binary\_to\_decimal(long int n)

{

int decimal = 0, i = 0, remainder;

while (n != 0)

{

remainder = n % 10;

n /= 10;

decimal += remainder \* pow(2, i);

++i;

}

return decimal;

}

int main()

{

long int n;

scanf("%ld", & n);

printf("%d",binary\_to\_decimal(n));

return 0;

}

**1 Coding Question (10 mark)**

Q.1. Program to find all possible permutations in which 'n' people can occupy 'r' seats in a theatre. (10 Marks)

N friends are planning to go to a movie. One among them suggested few movies and all others started to discuss and finally they selected a movie. One among them quickly booked their tickets online, to their surprise they are unable to select their seats. All of them got confused. Then anyhow, decided to go to the movie. They rushed to reach the theatre on time. Again, they are surprised that no one was there in the theatre. They are the only people about to watch the movie. There is 'r' number of seats in which, 'n' number persons should sit. In how many ways they can sit inside the theatre?

Given the number of people 'n' and the number of seats 'r' as input. The task is to find the different number of ways in which 'n' number of people can be seated in those 'r' number of seats.

For example,

Input:

Number of people: 5

Number of Rows: 3

Output:

The total number of ways in which 'n' people can be seated in 'r' seats = 60.

Calculation:

P(n,r) =P(5,3)

=5! /(5?3)! = 5! / ( 5 ? 3 )!

= 120 / 2 = 60

**Sample Input 1**

5

3

**Sample Output 1**

60

**Sample Input 2**

7

3

**Sample Output 2**

210

**Input Explanation**

Input consists of two integer value

First input will be number of people

Second input will be number of Rows

**Output Explanation**

Output consists of single integer value

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 12  1 | 11  4 | 9  3 | 6  2 | 4  6 |
| **Output** | 12 | 7920 | 504 | 30 | 360 |

**#Solution**

#include<stdio.h>

int fact(long int x)

{

long int f = 1, i;

for (i = 1; i <= x; i++)

{

f = f \* i;

}

return f;

}

int main()

{

long int n, r, p, temp;

long int num, den;

scanf("%ld",&r);

scanf("%ld",&n);

if (n < r)

{

temp = n;

n = r;

r = temp;

}

num = fact(n);

den = fact(n - r);

p = num / den;

printf("%ld", p);

}