Top of Form

Given two integers, join left-half of first number and right-half of second number. Both numbers will have even number of digits.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| joinHalfs(123456, 123456) | 123456 123456 | 123456 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

int joinHalfs(int num1, int num2) {

// Write your code here

}

int main() {

int m, n;

scanf("%d", &m);

scanf("%d", &n);

printf("%d\n", joinHalfs(m, n));

return 0;

}



**Question 2**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given an integer, find the middle digit. Given number will have odd number of digits.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| middleDigit(12345) | 12345 | 3 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

int middleDigit(int n) {

// Write your code here

}

int main() {

int n;

scanf("%d", &n);

printf("%d\n", middleDigit(n));

return 0;

}



**Question 3**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given an integer, compute number of digits.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| numberOfDigits(12345) | 12345 | 5 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

int numberOfDigits(int n) {

// Write your code here

}

int main() {

int n;

scanf("%d", &n);

printf("%d\n", numberOfDigits(n));

return 0;

}



**Question 4**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given an integer, find number of prime digits.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| numberOfPrimeDigits(12345) | 12345 | 3 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

#define TRUE 1

#define FALSE 0

// Given a digit

// return TRUE if prime otherwise

// return FALSE

int isPrime(int digit) {

switch (digit) {

case 2:

case 3:

case 5:

case 7:

return TRUE;

default:

return FALSE;

}

}

int numberOfPrimeDigits(int n) {



**Question 5**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given a number, find number of prime factors less than 10.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| numberOfPrimeFactors(12) | 12 | 2 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

int numberOfPrimeFactors(int n) {

// Write your code here

}

int main() {

int n;

scanf("%d", &n);

printf("%d\n", numberOfPrimeFactors(n));

return 0;

}



**Question 6**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given an integer, check if it is a palindrome in base 2 format. e.g: 5 in binary is 101 and it's palindrome.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| palindromeInBinary(5) | 5 | TRUE |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

#define TRUE 1

#define FALSE 0

// if n is palindrome in binary format

// return TRUE

// otherwise return FALSE

int palindromeInBinary(int n) {

// Write your code here

}

int main() {

int n, result;

scanf("%d", &n);

result = palindromeInBinary(n);

printf("%s\n", result ? "TRUE" : "FALSE");

return 0;

}



**Question 7**

Not complete

Marked out of 4.00

Flag question

**Question text**

Given two integers, M and N , remove all the digits from M which are in N. Both numbers will have digits from 1 to 9 only. e.g: M = 65631, N = 62373, the common digits are 6 and 3, so when we remove them from M, we will have 51.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| removeCommonDigits(65631, 62373) | 65631 62373 | 51 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

int removeCommonDigits(int m, int n) {

// Write your code here

}

int main() {

int m, n;

scanf("%d", &m);

scanf("%d", &n);

printf("%d\n", removeCommonDigits(m, n));

return 0;

}



**Question 8**

Not complete

Marked out of 4.00

Flag question

**Question text**

Given an integer, remove the repeating instance of digits. All digits in the input number will be from 1 to 9 only. e.g: 1234112 => 1234 since digits 1 and 2 are repeating, when we remove them, we get 1234.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| removeRepeatingDigits(1234112) | 1234112 | 1234 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

int removeRepeatingDigits(int n) {

// Write your code here

}

int main() {

int n;

scanf("%d", &n);

printf("%d\n", removeRepeatingDigits(n));

return 0;

}



**Question 9**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given an integer, compute sum of digits. If the result is not a single digit, repeat sum of digits until you get a single digit.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| repeatedSumOfDigits(12345) | 12345 | 6 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

int repeatedSumOfDigits(int n) {

// Write your code here

}

int main() {

int n;

scanf("%d", &n);

printf("%d\n", repeatedSumOfDigits(n));

return 0;

}



**Question 10**

Not complete

Marked out of 1.00

Flag question

**Question text**

Store name and CGPA of 'N' Students and list Names and CGPA of students who failed as per the CGPA.

Student is considered as Fail if CGPA is less than 5.

Number of students and details of all students must be taken as input.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 2  Sridhar  7.8  Manasa  4.0 | Manasa  4.0 |

Answer:(penalty regime: 0, 0, 0, 5, 10, 20, ... %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

#include<stdio.h>

struct std

{

char name[10];

float num;

}s[100];

int main()

{

int i;

float n;

scanf("%f",&n);

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

{

scanf("%s",s[i].name);

scanf("%f",&s[i].num);

}

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

{

if (s[i].num<5)

{

printf("%s\n",s[i].name);

printf("%.1f",s[i].num);



**Feedback**

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 2  Sridhar  7.8  Manasa  4.0 | Manasa  4.0 | Manasa  4.0 |  |
|  | 3  Sridhar  5.0  Manvi  8  Mangala  3.4 | Mangala  3.4 | Mangala  3.4 |  |

Run using the University of Canterbury's Jobe server. This is for initial testing only. Please set up your own Jobe server as soon as possible. See [here](https://github.com/trampgeek/moodle-qtype_coderunner/blob/master/Readme.md#sandbox-configuration).

Passed all tests!

**Question 11**

Not complete

Marked out of 1.00

Flag question

**Question text**

Store employee name and salary and display details of all employees whose salary is less than 10000

Take number of employees and details of employees as input.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 2  Ramana  8000  Ravi  12000 | Ramana  8000 |

Answer:(penalty regime: 0, 0, 0, 5, 10, 20, ... %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

#include<stdio.h>

struct std

{

char name[10];

int num;

}s[100];

int main()

{

int i,n;

scanf("%d",&n);

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

{

scanf("%s",s[i].name);

scanf("%d",&s[i].num);

}

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

{

if (s[i].num<10000)

{

printf("%s\n",s[i].name);

printf("%d",s[i].num);

}



**Feedback**

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 2  Ramana  8000  Ravi  12000 | Ramana  8000 | Ramana  8000 |  |
|  | 3  Sita  10000  Rama  12000  Lakshmana  7000 | Lakshmana  7000 | Lakshmana  7000 |  |

Run using the University of Canterbury's Jobe server. This is for initial testing only. Please set up your own Jobe server as soon as possible. See [here](https://github.com/trampgeek/moodle-qtype_coderunner/blob/master/Readme.md#sandbox-configuration).

Passed all tests!

**Question 12**

Not complete

Marked out of 1.00

Flag question

**Question text**

Store person's name and age and display details of all persons whose age is greater than 60.

Take number of persons and details of persons as input.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 2  Anudeep  12  Ramana  75 | Ramana  75 |

Answer:(penalty regime: 0, 0, 0, 5, 10, 20, ... %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

#include<stdio.h>

struct std

{

char name[10];

int num;

}s[100];

int main()

{

int i,n;

scanf("%d",&n);

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

{

scanf("%s",s[i].name);

scanf("%d",&s[i].num);

}

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

{

if (s[i].num>60){

printf("%s\n",s[i].name);

printf("%d",s[i].num);

}



**Feedback**

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 2  Anudeep  12  Ramana  75 | Ramana  75 | Ramana  75 |  |
|  | 3  Sita  60  Ramana  75  Ravi  40 | Ramana  75 | Ramana  75 |  |

Run using the University of Canterbury's Jobe server. This is for initial testing only. Please set up your own Jobe server as soon as possible. See [here](https://github.com/trampgeek/moodle-qtype_coderunner/blob/master/Readme.md#sandbox-configuration).

Passed all tests!

**Question 13**

Not complete

Marked out of 1.00

Flag question

**Question text**

Two characters B and L are to be displayed alternatively for the given numbers of rows and columns.

Character, rows and columns are to be accepted as input.

**For example:**

| **Input** | **Result** |
| --- | --- |
| B 4 3 | B L B  L B L  B L B  L B L |
| L 6 6 | L B L B L B  L B L B L B  L B L B L B  L B L B L B  L B L B L B  L B L B L B |

Answer:(penalty regime: 0 %)



1



**Question 14**

Not complete

Marked out of 1.00

Flag question

**Question text**

Write a C program to find position of greatest element in a m\*n matrix. If more elements are there with same value, list all the positions

Take rows and columns of the matrix and elements of the matrix as inputs.

elements to be stored are of data type float.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 5 5  32 24 65 42 23  18 92 89 96 90  34 35 36 37 38  40 32 90.5 91.5 83  83.5 78.3 90.5 98.5 72.5 | 4 3 |
| 3 4  23 32 25 24.5  35 26 78 90  90 76 72 89 | 1 3  2 0 |

Answer:(penalty regime: 0 %)



1



**Question 15**

Not complete

Marked out of 1.00

Flag question

**Question text**

Write a C Program which accepts m\*n matrix and displays minimum element in a given column

**For example:**

| **Input** | **Result** |
| --- | --- |
| 2 3  2  1 2 3  4 5 6 | 3 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

/\*4. Given a two-dimensional array of integers and a row index, return the largest element in

that row\*/

#include<stdio.h>

int main()

{

int a[10][10],i,j,rows,cols,max,lag;

scanf("%d%d",&rows,&cols);

scanf("%d",&lag);

for(i=0;i<rows;i++)

{

for(j=0;j<cols;j++)

{

scanf("%d",&a[i][j]);

}

}

max = a[0][lag];

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

{



**Feedback**

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 2 3  2  1 2 3  4 5 6 | 3 | 0 |  |
|  | 4 5  3  -9 -5 -2 -3 6  -6 -4 3 2 1  9 6 5 4 3  3 2 4 6 7 | -3 | -3 |  |

Run using the University of Canterbury's Jobe server. This is for initial testing only. Please set up your own Jobe server as soon as possible. See [here](https://github.com/trampgeek/moodle-qtype_coderunner/blob/master/Readme.md#sandbox-configuration).

**Question 16**

Not complete

Marked out of 1.00

Flag question

**Question text**

Write a C Program which accepts m\*n matrix and displays maximum element in a given column

**For example:**

| **Input** | **Result** |
| --- | --- |
| 2 3  2  1 2 3  4 5 6 | 6 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

/\*4. Given a two-dimensional array of integers and a row index, return the largest element in

that row\*/

#include<stdio.h>

int main()

{

int a[10][10],i,j,rows,cols,max,lag;

scanf("%d%d",&rows,&cols);

scanf("%d",&lag);

for(i=0;i<rows;i++)

{

for(j=0;j<cols;j++)

{

scanf("%d",&a[i][j]);

}

}

max = a[0][lag];

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

{



**Feedback**

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 2 3  2  1 2 3  4 5 6 | 6 | 6 |  |
|  | 4 5  3  -9 -5 -2 -3 6  -6 -4 3 2 1  9 6 5 4 3  3 2 4 6 7 | 6 | 6 |  |

Run using the University of Canterbury's Jobe server. This is for initial testing only. Please set up your own Jobe server as soon as possible. See [here](https://github.com/trampgeek/moodle-qtype_coderunner/blob/master/Readme.md#sandbox-configuration).

Passed all tests!

**Question 17**

Not complete

Marked out of 1.00

Flag question

**Question text**

Write a program to swap i and j column values in a given m\*n matrix.

Size of the array, both columns which are to be swapped and elements of the matrix are given as inputs.

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4 6  0 5  11 12 13 14 15 16  21 22 23 24 25 26  31 32 33 34 35 36  41 42 43 44 45 46 | 16 12 13 14 15 11  26 22 23 24 25 21  36 32 33 34 35 31  46 42 43 44 45 41 |

Answer:(penalty regime: 0 %)



1



**Question 18**

Not complete

Marked out of 1.00

Flag question

**Question text**

Find factorial of a given integer N>=0. If N<0, print "Invalid Input".

Example:

Input:

0

Output

1

Input:

5

Output:

120

Input: -8

Output:

Invalid Input

**For example:**

| **Input** | **Result** |
| --- | --- |
| 0 | 1 |

Answer:(penalty regime: 10, 20, ... %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

#include<stdio.h>

int main()

{

int n,i,fact=1;

scanf("%d",&n);

if(n>=0)

{

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

{

fact=fact\*i;

}

printf("%d",fact);

}

else

{

printf("Invalid Input");

}

}



**Feedback**

|  | **Input** | **Expected** | **Got** |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | 1 |  |
|  | -9 | Invalid Input | Invalid Input |  |
|  | 5 | 120 | 120 |  |
|  | 1 | 1 | 1 |  |
|  | 2 | 2 | 2 |  |

Run using the University of Canterbury's Jobe server. This is for initial testing only. Please set up your own Jobe server as soon as possible. See [here](https://github.com/trampgeek/moodle-qtype_coderunner/blob/master/Readme.md#sandbox-configuration).

Passed all tests!

**Question 19**

Not complete

Marked out of 1.00

Flag question

**Question text**

Check whether a number N > 0 is palindrome or not. If N<=0, print Invalid Input

Input:

131

Output:

Palindrome

Input:   
123  
OUPUT:

Not a palindrome

Input:

-9

Output:

Invalid Input

**For example:**

| **Input** | **Result** |
| --- | --- |
| 131 | Palindrome |
| 123 | Not a palindrome |

Answer:(penalty regime: 10, 20, ... %)



1



**Question 20**

Not complete

Marked out of 1.00

Flag question

**Question text**

Write a C program to find whether a given number is perfect number or not. If N<=0, print as Invalid Number.

Input:

6

Output:

Perfect Number

Input:

12

Output:

Not a perfect number

**For example:**

| **Input** | **Result** |
| --- | --- |
| 6 | Perfect Number |

Answer:(penalty regime: 10, 20, ... %)



1



**Question 21**

Not complete

Marked out of 1.00

Flag question

**Question text**

Print the following pattern as per the level.

Input:

4

Output:

\*

\* \*

\* \* \*

\* \* \* \*

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4 | \*  \* \*  \* \* \*  \* \* \* \* |

Answer:(penalty regime: 10, 20, ... %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

#include<stdio.h>

int main()

{

int n,i,j;

scanf("%d",&n);

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

{

printf("\n");

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

{

printf("\* ");

}

}

}



**Feedback**

**Question 22**

Not complete

Marked out of 1.00

Flag question

**Question text**

Print the following pattern as per the given level.

Input:

4

A

B B

C C C

D D D D

**For example:**

| **Input** | **Result** |
| --- | --- |
| 4 | A  B B  C C C  D D D D |

Answer:(penalty regime: 10, 20, ... %)



1



**Question 23**

Not complete

Marked out of 1.00

Flag question

**Question text**

Print the following pattern, based on the given level:

input:

4

1

2 3

4 5 6

7 8 9 10

**For example:**

| **Input** | **Result** |
| --- | --- |
| 3 | 1  2 3  4 5 6 |

Answer:(penalty regime: 10, 20, ... %)



1



**Question 24**

Not complete

Marked out of 1.00

Flag question

**Question text**

Print whether a given number is prime or not.

Input:

2

Output:

Prime number

Input:

12

Output:

Not a prime number

**For example:**

| **Input** | **Result** |
| --- | --- |
| 2 | Prime number |

Answer:(penalty regime: 10, 20, ... %)



1



**Question 25**

Not complete

Marked out of 1.00

Flag question

**Question text**

Find GCD of two numbers

Input:

6 7

Output

1

**For example:**

| **Input** | **Result** |
| --- | --- |
| 6 26 | 2 |

Answer:(penalty regime: 10, 20, ... %)



1



**Question 26**

Not complete

Marked out of 1.00

Flag question

**Question text**

Write a C program to count for a particular character in a given string.

Example:

Input:

Jayasri

a

Output:

2

**For example:**

| **Input** | **Result** |
| --- | --- |
| Aaradhya  a | 4 |
| Rama killed Ravana  R | 2 |

Answer:(penalty regime: 0, 0, 0, 5, 10, 20, ... %)



1



**Question 27**

Not complete

Marked out of 1.00

Flag question

**Question text**

Write a C program to find number of words in a given sentence.

Example:

Input:

Be happy and cheerful!

Output:

4

**For example:**

| **Input** | **Result** |
| --- | --- |
| Be happy and cheerful! | 4 |
| A B C | 3 |

Answer:(penalty regime: 0, 0, 0, 5, 10, 20, ... %)



1





Bottom of Form

[◄ Module 5 (Content)](https://learn.gitam.edu/mod/resource/view.php?id=331842&forceview=1)

Top of Form

Jump to...

Top of Form

Given an integer, reverse the order of least significant 3 digits.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| reverseLast3(12345) | 12345 | 12543 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

int reverseLast3(int n) {

// Write your code here

}

int main() {

int n;

scanf("%d", &n);

printf("%d\n", reverseLast3(n));

return 0;

}



**Question 29**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given two numbers, return TRUE if they both have the same sign. Otherwise return FALSE.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| joinHalfs(23, 25) | 23 25 | TRUE |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

#define TRUE 1

#define FALSE 0

// return TRUE or FALSE

int sameSign(int num1, int num2) {

// Write your code here

}

int main() {

int m, n, result;

scanf("%d", &m);

scanf("%d", &n);

result = sameSign(m, n);

printf("%s\n", result ? "TRUE" : "FALSE");

return 0;

}



**Question 30**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given an integer, compute sum of cubes of least-significant-digit and most-significant-digit.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| sumOfCubesOfLSDAndMSD(12345) | 12345 | 126 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

int sumOfCubesOfLSDAndMSD(int n) {

// Write your code here

}

int main() {

int n;

scanf("%d", &n);

printf("%d\n", sumOfCubesOfLSDAndMSD(n));

return 0;

}



**Question 31**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given an integer, compute sum of each digit raised to its position power. e.g: 345 => (3^1 + 4^2 + 5^3) => (3 + 16 + 125) => 144. Here symbol ^ represents power.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| sumOfDigitsRaisedToPosPower(345) | 345 | 144 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

// computes a power b

int power(int a, int b) {

int result;

if (a == 0) {

result = 0;

} else {

result = 1;

}

while (b > 0) {

result \*= a;

b = b - 1;

}

return result;

}

int sumOfDigitsRaisedToPosPower(int n) {

// Write your code here

// Use the above function

}

int main() {



**Question 32**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given an integer, compute sum of factorial of digits.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| sumOfFactorialOfDigits(12345) | 12345 | 153 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

// factorial of a given digit

int factorial(int digit) {

int fact[10] = {1, 1, 2};

for (int n = 3; n < 10; n++) {

fact[n] = fact[n-1] \* n;

}

return fact[digit];

}

int sumOfFactorialOfDigits(int n) {

// Write your code here

// Use the above function

}

int main() {

int n;

scanf("%d", &n);

printf("%d\n", sumOfFactorialOfDigits(n));

return 0;

}



**Question 33**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given an integer, compute sum of least-significant-digit and most-significant-digit.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| sumOfLSDAndMSD(12345) | 12345 | 6 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

int sumOfLSDAndMSD(int n) {

// Write your code here

}

int main() {

int n;

scanf("%d", &n);

printf("%d\n", sumOfLSDAndMSD(n));

return 0;

}



**Question 34**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given two integers, compute sum of middle digits. Both numbers will have odd number of digits.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| sumOfMiddleDigits(214, 23434) | 214 23434 | 5 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

int sumOfMiddleDigits(int num1, int num2) {

// Write your code here

}

int main() {

int m, n;

scanf("%d", &m);

scanf("%d", &n);

printf("%d\n", sumOfMiddleDigits(m, n));

return 0;

}



**Question 35**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given two integers, compute sum of most-significant-digit of each number.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| sumOfMSD(2342, 234234) | 2342 234234 | 4 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

int sumOfMSD(int num1, int num2) {

// Write your code here

}

int main() {

int m, n;

scanf("%d", &m);

scanf("%d", &n);

printf("%d\n", sumOfMSD(m, n));

return 0;

}



**Question 36**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given an integer, compute sum of squares of least-significant-digit and most-significant-digit.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| sumOfSquareOfLSDAndMSD(12345) | 12345 | 26 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

int sumOfSquareOfLSDAndMSD(int n) {

// Write your code here

}

int main() {

int n;

scanf("%d", &n);

printf("%d\n", sumOfSquareOfLSDAndMSD(n));

return 0;

}



**Question 37**

Not complete

Marked out of 2.00

Flag question

**Question text**

Given three integers, compute sum of squares of most-significant-digit of each number.  
Write your approach/logic/algorithm in the comments area (Lines 4, 5, 6, 7, 8). And implement the function.

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| sumOfSquaresOfMSDs(2342, 234234, 23) | 2342 234234 23 | 12 |

Answer:(penalty regime: 0 %)



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

#include <stdio.h>

// Explain your logic here as comments

//

//

//

//

//

// return most significant digit of n

int msd(int n) {

while (n >= 10) {

n /= 10;

}

return n;

}

int sumOfSquaresOfMSDs(int num1, int num2, int num3) {

// Write your code here

// use the above function to compute this result

}

int main() {

int l, m, n;

scanf("%d", &l);

scanf("%d", &m);

scanf("%d", &n);

printf("%d\n", sumOfSquaresOfMSDs(l, m, n));





Bottom of Form

[◄ Module 5 (Content)](https://learn.gitam.edu/mod/resource/view.php?id=331842&forceview=1)

Top of Form

Jump to...

Bottom of Form

Bottom of Form