[Skip to main content](http://rajalakshmiinstitutions.net/moodle/mod/quiz/review.php?attempt=2163&cmid=209&maincontent)

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Friday, 29 September 2024, 1:27 PM

**State** Finished

**Completed on** Friday, 29 September 2024, 2:25 PM

**Time taken** 57 mins 54 secs



## **Question 1**

Correct

Marked out of 3.00



**Question text**

**Objective**

This is a simple challenge to help you practice printing to stdout.



We're starting out by printing the most famous computing phrase of all time! In the editor below, use either printf or cout to print the string ***Hello, World!*** to stdout.

**Input Format**

You do not need to read any input in this challenge.

**Output Format**

Print ***Hello, World!*** to stdout.

**Sample Output**



| 1 | #include<stdio.h> |  | 1 |
| --- | --- | --- | --- |
| 2  3  4  5 | int main() {  printf("Hello, World!"); return 0;  } |  |

Hello, World! Answer:(penalty regime: 0 %)



**Feedback**

**Expected Got**

Hello, World! Hello, World!

Passed all tests!

## **Question 2**

Correct

Marked out of 5.00 

**Question text Objective**

This challenge will help you to learn how to take a character, a string and a sentence as input in C.

To take a single character ***ch*** as input, you can use scanf("%c", &ch); and printf("%c", ch) writes a character specified by the argument char to stdout:

char ch; scanf("%c", &ch);

printf("%c", ch);

This piece of code prints the character ***ch***.

**Task**

You have to print the character, ***ch***.

**Input Format**

Take a character, ***ch*** as input.

**Output Format**

Print the character, ***ch***.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 2 |
| --- | --- | --- | --- |
| 2  3  4  5 | int main() {  char ch; scanf("%c",&ch);  printf("%c",ch); |  |

| 6  7 | return 0;  } |  |  |
| --- | --- | --- | --- |
|  | | | |

**Feedback**

**Input Expected Got**

C C C

Passed all tests!

## **Question 3**

Correct

Marked out of 7.00

**Question text Objective**

The fundamental data types in c are int, float and char. Today, we're discussing int and float data types.

The printf() function prints the given statement to the console. The syntax is printf("format string",argument\_list);. In the function, if we are using an integer, character, string or float as argument, then in the format string we have to write %d (integer), %c (character), %s (string), %f (float) respectively.

The scanf() function reads the input data from the console. The syntax is scanf("format string",argument\_list);. For ex:

The scanf("%d",&number) statement reads integer number from the console and stores the given value in variable ***number***.

To input two integers separated by a space on a single line, the command is scanf("%d %d", &n, &m), where ***n*** and ***m*** are the two integers.

**Task**

Your task is to take two numbers of [int data type](http://rajalakshmiinstitutions.net/moodle/mod/quiz/view.php?id=158), two numbers of float data type as input and output their sum:

1. Declare ***4*** variables: two of type int and two of type float.
2. Read ***2*** lines of input from stdin (according to the sequence given in the 'Input Format' section below) and initialize your ***4*** variables.
3. Use the ***+*** and ***-*** operator to perform the following operations:
   * Print the sum and difference of two int variable on a new line.

3

* + Print the sum and difference of two float variable rounded to one decimal place on a new line.

**Input Format**

The first line contains two integers.

The second line contains two floating point numbers.

**Constraints**

* ***1 ≤ integer variables ≤ 104***
* ***1 ≤ float variables ≤ 104***

**Output Format**

Print the sum and difference of both integers separated by a space on the first line, and the sum and difference of both float (scaled to ***1*** decimal place) separated by a space on the second line.

**Sample Input**

10 4

4.0 2.0

**Sample Output**

14 6

6.0 2.0

**Explanation**

When we sum the integers ***10*** and ***4***, we get the integer ***14***. When we subtract the second number ***4*** from the first number ***10***, we get ***6*** as their difference.

When we sum the floating-point numbers ***4.0*** and ***2.0***, we get ***6.0***. When we subtract the second number ***2.0*** from the first number ***4.0***, we get ***2.0*** as their difference.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 4 |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8 | int main() {  int a,b; float c,d;  scanf("%d%d",&a,&b);  scanf("%f%f",&c,&d);  printf("%d %d \n",a+b,a-b);  printf("%.1f %.1f",c+d,c-d); |  |

| 9  10  11 | return 0;  } |  |  |
| --- | --- | --- | --- |
|  | | | |

**Feedback**

**Input Expected Got**

10 4

14 6

14 6

4.0 2.0 6.0 2.0 6.0 2.0

20 8

28 12

28 12

8.0 4.0 12.0 4.0 12.0 4.0

Passed all tests!

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Tuesday, 3 October 2024, 8:31 AM

**State** Finished

**Completed on** Tuesday, 3 October 2024, 8:44 AM

**Time taken** 13 mins 28 secs

## **Question 1**

Correct

Marked out of 3.00

**Question text**

Write a program to input a name (as a single character) and marks of three tests as m1, m2, and m3 of a student considering all the three marks have been given in integer format.

Now, you need to calculate the average of the given marks and print it along with the name as mentioned in the output format section.

All the test marks are in integers and hence calculate the average in integer as well. That is, you need to print the integer part of the average only and neglect the decimal part.

Input format :

Line 1 : Name(Single character)

Line 2 : Marks scored in the 3 tests separated by single space.

Output format :

First line of output prints the name of the student. Second line of the output prints the average mark.

Constraints

Marks for each student lie in the range 0 to 100 (both inclusive)

Sample Input 1 :

A

3 4 6

Sample Output 1 :

A 4

Sample Input 2 :

T

7 3 8

Sample Output 2 :

T 6

Answer:(penalty regime: 0 %)

| 1 | #include <stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12 | int main() { char name; int m1,m2,m3; int average;  scanf("%c",&name);  scanf("%d %d %d",&m1,&m2,&m3); average=(m1+m2+m3)/3; printf("%c\n",name); printf("%d\n",average);  return 0;  } |  |
|  | | | |

**Feedback**

| **Input** | **Expected** | **Got** |
| --- | --- | --- |
| A | A | A |
| 3 4 6 | 4 | 4 |
| T | T | T |
| 7 3 8 | 6 | 6 |
| R | R | R |
| 0 100 99 | 66 | 66 |

Passed all tests!

## **Question 2**

Correct

Marked out of 5.00

**Question text**

Some *C* data types, their format specifiers, and their most common bit widths are as follows:

* *Int ("%d"):* 32 Bit integer
* *Long ("%ld"):* 64 bit integer
* *Char ("%c"):* Character type
* *Float ("%f"):* 32 bit real value
* *Double ("%lf"):* 64 bit real value



**Reading**

To read a data type, use the following syntax:

scanf("`format\_specifier`", &val)

For example, to read a *character* followed by a *double*:

char ch; double d;

scanf("%c %lf", &ch, &d);

For the moment, we can ignore the spacing between format specifiers.



**Printing**

To print a data type, use the following syntax:

printf("`format\_specifier`", val)

For example, to print a *character* followed by a *double*:

char ch = 'd'; double d = 234.432;

printf("%c %lf", ch, d);

**Note:** You can also use *cin* and *cout* instead of *scanf* and *printf*; however, if you are taking a million numbers as input and printing a million lines, it is faster to use *scanf* and *printf*.

**Input Format**

Input consists of the following space-separated values: *int*, *long*, *char*, *float*, and *double*, respectively.

**Output Format**

Print each element on a new line in the same order it was received as input. Note that the floating point value should be correct up to 3 decimal places and the double to 9 decimal places.

**Sample Input**

3 12345678912345 a 334.23 14049.30493

**Sample Output**

3

12345678912345

a 334.230

14049.304930000

**Explanation**

Print *int* ***3***,

followed by *long* ***12345678912345***, followed by *char* ***a***,

followed by *float* ***334.23***,

followed by *double* ***14049.30493***.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13  14  15 | int main() { int a; long b; char c; float d; double e;  scanf("%d %ld %c %f %lf",&a,&b,&c,&d,&e); printf("%d\n",a);  printf("%ld\n",b);  printf("%c\n",c);  printf("%.3f\n",d);  printf("%.9lf\n",e); return 0;  } |  |
|  | | | |

**Feedback**

**Input Expected Got**

3

12345678912345

3 12345678912345 a 334.23 14049.30493 a

334.230

3

12345678912345

a 334.230

14049.304930000 14049.304930000

Passed all tests!

## **Question 3**

Correct

Marked out of 7.00

**Question text**

Write a program to print the [ASCII value](http://rajalakshmiinstitutions.net/moodle/mod/quiz/view.php?id=8) and the two adjacent characters of the given character.

Input

E

Output

69

D F

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 9 |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9 | int main() {  char ch; scanf("%c",&ch); int a=(int)ch; printf("%d\n",a); char b=ch-1;  char c=ch+1;  printf("%c %c \n",b,c); |  |

| 10  11  12 | return 0;  } |  |  |
| --- | --- | --- | --- |
|  | | | |

**Feedback**

**Input Expected Got**

E 69 69

D F D F

Passed all tests!

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Friday, 10 November 2024, 1:03 PM

**State** Finished

**Completed on** Friday, 10 November 2024, 1:18 PM

**Time taken** 14 mins 25 secs

## **Question 1**

Correct

Marked out of 3.00

**Question text**

Many people think about their height in feet and inches, even in some countries that primarily use the metric system. Write a program that reads a number of feet from the user, followed by a number of inches. Once these values are read, your program should compute and display the equivalent number of centimeters.

Hint:

One foot is 12 inches.

One inch is 2.54 centimeters. Input Format

First line,read the number of feet. Second line, read the number of inches. Output Format

In one line print the height in centimeters.

Note: All of the values should be displayed using two decimal places. Sample Input 1

5 6

Sample Output 1

167.64

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10 | int main()  {  int a; int b;  scanf("%d",&a);  scanf("%d",&b); printf("%.2f",(a\*12\*2.54)+(b\*2.54)); return 0;  } |  |
|  | | | |

**Input Expected Got**

5 167.64 167.64

6

Passed all tests!

## **Question 2**

Correct

Marked out of 5.00

**Question text**

Create a program that reads two integers, a and b, from the user. Your program should compute and display: • The sum of a and b • The difference when b is subtracted from a • The product of a and b • The quotient when a is divided by b • The remainder when a is divided by b

Input Format

First line, read the first number. Second line, read the second number. Output Format

First line, print the sum of a and b

Second line, print the difference when b is subtracted from a Third line, print the product of a and b

Fourth line, print the quotient when a is divided by b Fifth line, print the remainder when a is divided by b Sample

Input 1 100 6 Sample Output 106 94 600 16 4

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12 | int main()  {  int a,b;  scanf("%d %d",&a,&b);  printf("%d\n",a+b);  printf("%d\n",a-b);  printf("%d\n",a\*b);  printf("%d\n",a/b);  printf("%d\n",a%b); return 0;  } |  |
|  | | | |

**Feedback**

**Input Expected Got**

100

106

94

600

106

94

600 12

6 16 16

4 4

Passed all tests!

## **Question 3**

Correct

Marked out of 7.00

**Question text**

A bakery sells loaves of bread for $3.49 each. Day old bread is discounted by 60 percent. Write a program that begins by reading the number of loaves of day old bread being purchased from the user. Then your program should display the regular price for the bread, the discount because it is a day old, and the total price. Each of these amounts should be displayed on its own line with an appropriate label. All of the values should be displayed using two decimal places.

Input Format

Read the number of day old loaves. Output Format

First line, print Regular price: price Second line, print Discount: discount Third line, print Total: total

Note: All of the values should be displayed using two decimal places. Sample Input 1

10

Sample Output 1

Regular price: 34.90

Discount: 20.94

Total: 13.96

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8 | int main()  {  int a; scanf("%d",&a);  printf("Regular price: %.2f\nDiscount: %.2f\nTotal: %.2f\n",a\*3.49,a\*3.49\*0.6,a\*3.49\*0. return 0;  } | 4); |
|  | | | |

**Feedback**

**Input Expected Got**

Regular price: 34.90 Regular price: 34.90

10 Discount: 20.94

Total: 13.96

Discount: 20.94

Total: 13.96

13



REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Monday, 13 November 2024, 8:29 PM

**State** Finished

**Completed on** Monday, 13 November 2024, 9:32 PM

**Time taken** 1 hour 2 mins

## **Question 1**

Correct

Marked out of 3.00

**Question text**

Goki recently had a breakup, so he wants to have some more friends in his life. Goki has N people who he can be friends with, so he decides to choose among them according to their skills set Yi(1<=i<=n). He wants atleast X skills in his friends. Help Goki find his friends.

INPUT

First line contains a single integer X - denoting the minimum skill required to be Goki's friend. Next line contains one integer Y - denoting the skill of the person

.

OUTPUT

Print if he can be friend with Goki. 'YES' (without quotes) if he can be friends with Goki else 'NO' (without quotes).



CONSTRAINTS 1<=N<=1000000

1<=X,Y<=1000000 SAMPLE INPUT 1

100 110

SAMPLE OUTPUT 1 YES

SAMPLE INPUT 2

100 90

SAMPLE OUTPUT 2 NO

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | int main() |  |  |
| 3 | { |  |
| 4 | int x,y; |  |
| 5 | scanf("%d %d",&x,&y); |  |
| 6 |  |  |
| 7 | if(y>=x) |  |
| 8 | { |  |
| 9 | printf("YES"); |  |
| 10 | } |  |
| 11 | else |  |
| 12 | { |  |
| 13 | printf("NO"); |  |
| 14 |  |  |
| 15 | } |  |
| 16 |  | 14 |

| 178 | } |  |  |
| --- | --- | --- | --- |
|  | | | |

**Feedback**

**Input Expected Got**

100

110

YES YES

100

90

NO NO

Passed all tests!

## **Question 2**

Correct

Marked out of 5.00

**Question text**

Before the outbreak of corona virus to the world, a meeting happened in a room in Wuhan. A person who attended that meeting had COVID-19 and no one in the room knew about it! So everyone started shaking hands with everyone else in the room as a gesture of respect and after meeting unfortunately everyone got infected! Given the fact that any two persons shake hand exactly once, Can you tell the total count of handshakes happened in that meeting? Say no to shakehands. Regularly wash your hands. Stay Safe.

Input Format

Read an integer N,the total number of people attended that meeting. Output Format

Print the number of handshakes. Constraints

0 < N < 106

SAMPLE INPUT 1

1

SAMPLE OUTPUT 0

SAMPLE INPUT 2

2

SAMPLE OUTPUT 2

1

Explanation Case 1: The lonely board member shakes no hands, hence 0. Case 2: There are 2 board members, 1 handshake takes place.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 15 |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8 | int main()  {  int a,b; scanf("%d",&a); b=a\*(a-1)/2; printf("%d",b);  } |  |



**Feedback**

**Input Expected Got**

| 1 | 0 | 0 |
| --- | --- | --- |
| 2 | 1 | 1 |

Passed all tests!

## **Question 3**

Correct

Marked out of 7.00

**Question text**

In our school days, all of us have enjoyed the Games period. Raghav loves to play cricket and is Captain of his team. He always wanted to win all cricket matches. But only one last Games period is left in school now. After that he will pass out from school. So, this match is very important to him. He does not want to lose it. So he has done a lot of planning to make sure his teams wins. He is worried about only one opponent - Jatin, who is very good batsman. Raghav has figured out 3 types of bowling techniques, that could be most beneficial for dismissing Jatin. He has given points to each of the 3 techniques. You need to tell him which is the maximum point value, so that Raghav can select best technique. 3 numbers are given in input. Output the maximum of these numbers.

Input:

Three space separated integers. Output:

Maximum integer value SAMPLE INPUT

8 6 1

SAMPLE OUTPUT 8



| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | int main() |  |  |
| 3 | { |  |
| 4 | int a,b,c; |  |
| 5 |  |  |
| 6 | scanf("%d %d %d",&a,&b,&c); |  |
| 7 | int x=a; |  |
| 8 | if(b>x) |  |
| 9 | { |  |
| 10 | x=b; |  |
| 11 | } |  |
| 12 | if(c>x) |  |
| 13 | { |  |
| 14 | x=c; |  |
| 15 | } |  |
| 16 | printf("%d\n",x); |  |
| 17 | return 0; |  |
| 18 | } | 16 |

Explanation Out of given numbers, 8 is maximum. Answer:(penalty regime: 0 %)



**Feedback**

**Input Expected Got**

81 26 15 81 81

Passed all tests!

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Tuesday, 15 November 2024, 8:56 AM

**State** Finished

**Completed on** Tuesday, 15 November 2024, 9:08 AM

**Time taken** 11 mins 37 secs

## **Question 1**

Correct

Marked out of 3.00

**Question text**

Write a program to read two integer values and print true if both the numbers end with the same digit, otherwise print false. Example: If 698 and 768 are given, program should print true as they both end with 8. Sample Input 1 25 53 Sample Output 1 false Sample Input 2 27 77 Sample Output 2 true

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11 | int main() {  int x,y;  scanf("%d %d",&x,&y); if(x%10==y%10) {  printf("true"); } else {  printf("false");  }  } |  |
|  | | | |

**Feedback**

**Input Expected Got**

25 53 false false

27 77 true true

Passed all tests!

## **Question 2**

Correct

Marked out of 5.00

**Question text Objective**

In this challenge, we're getting started with conditional statements.

**Task**

Given an integer, ***n***, perform the following conditional actions:

* If ***n*** is odd, print Weird
* If ***n*** is even and in the inclusive range of ***2*** to ***5***, print ***Not Weird***
* If ***n*** is even and in the inclusive range of ***6*** to ***20***, print ***Weird***
* If ***n*** is even and greater than ***20***, print ***Not Weird***

Complete the stub code provided in your editor to print whether or not ***n*** is weird.

**Input Format**

A single line containing a positive integer, ***n***.

**Constraints**

* 1 < n < 100

**Output Format**

Print Weird if the number is weird; otherwise, print Not Weird.

**Sample Input 0**

3

**Sample Output 0**

Weird

**Sample Input 1**

24

**Sample Output 1**

Not Weird

**Explanation**

*Sample Case 0:* ***n = 3***

***n*** is odd and odd numbers are weird, so we print ***Weird***.

*Sample Case 1:* ***n = 24***

***n > 20*** and ***n*** is even, so it isn't weird. Thus, we print ***Not Weird***.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13  14  15 | int main() { int n;  scanf("%d",&n); if(n%2==0) {  if(n>=2 && n<=5) { printf("Not Weird"); }  if(n>=6 && n<=20) { printf("Weird"); } if(n>20) {  printf("Not Weird"); } }  else {  printf("Weird"); }  } |  |
|  | | | |

**Feedback**

**Input Expected Got**

3 Weird Weird

24 Not Weird Not Weird

Passed all tests!

## **Question 3**

Correct

Marked out of 7.00

**Question text**

Three numbers form a Pythagorean triple if the sum of squares of two numbers is equal to the square of the third. For example, 3, 5 and 4 form a Pythagorean triple, since 3\*3 + 4\*4 = 25 = 5\*5 You are given three integers, a, b, and c.

They need not be given in increasing order. If they form a Pythagorean triple, then print "yes", otherwise, print "no". Please note that the output message is in small letters. Sample Input 1 3 5 4 Sample Output 1 yes Sample Input 2 5 8 2 Sample Output 2 no

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 20 |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13  14 | int main() {  int a,b,c; scanf("%d%d%d",&a,&b,&c); if(a\*a+b\*b==c\*c) {  printf("yes"); } else if(a\*a+c\*c==b\*b) {  printf("yes"); } else if(b\*b+c\*c==a\*a) {  printf("yes"); } else {  printf("no"); } return 0;  } |  |



**Feedback**

**Input Expected Got**

| 3 |  | |
| --- | --- | --- |
| 5 | yes | yes |
| 4 |  |  |
| 5 |  |  |
| 8 | no | no |
| 2 |  |  |

Passed all tests!

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Sunday, 18 November 2024, 10:33 PM

**State** Finished

**Completed on** Sunday, 18 November 2024, 10:51 PM

**Time taken** 18 mins 8 secs

## **Question 1**

Correct

Marked out of 3.00

**Question text**

Write a program that determines the name of a shape from its number of sides. Read the number of sides from the user and then report the appropriate name as part of a meaningful message. Your program should support shapes with anywhere from 3 up to (and including) 10 sides. If a number of sides outside of this range is entered then your program should display an appropriate error message.

Sample Input 1

3

Sample Output 1

Triangle

Sample Input 2

7

Sample Output 2

Heptagon

Sample Input 3

11

Sample Output 3

The number of sides is not supported.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 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  32  33  34  35  36  37  38  39  40  41  42 | int main()  {  int n; scanf("%d",&n); if(n==3)  {  printf("Triangle");  }  else if(n==4)  {  printf("Square");  }  else if(n==5)  {  printf("Pentagon");  }  else if(n==6)  {  printf("Hexagon");  }  else if(n==7)  {  printf("Heptagon");  }  else if(n==8)  {  printf("Octagon");  }  else if(n==9)  {  printf("Nonagon");  }  else if(n==10)  {  printf("Decagon");  }  else  {  printf("The number of sides is not supported.");  }  } |  |
|  | | | |

**Feedback**

**Input Expected Got**

3 Triangle Triangle

7 Heptagon Heptagon

11 The number of sides is not supported. The number of sides is not supported.

Passed all tests!

## **Question 2**

Correct

Marked out of 5.00

**Question text**

The Chinese zodiac assigns animals to years in a 12-year cycle. One 12-year cycle is shown in the table below. The pattern repeats from there, with 2012 being another year of the Dragon, and 1999 being another year of the Hare.

Year Animal

| 2000 | Dragon |
| --- | --- |
| 2001 | Snake |
| 2002 | Horse |
| 2003 | Sheep |
| 2004 | Monkey |
| 2005 | Rooster |
| 2006 | Dog |
| 2007 | Pig |
| 2008 | Rat |
| 2009 | Ox |
| 2010 | Tiger |
| 2011 | Hare |



Write a program that reads a year from the user and displays the animal associated with that year. Your program should work correctly for any year greater than or equal to zero, not just the ones listed in the table.

Sample Input 1

2004

Sample Output 1

Monkey

Sample Input 2

2010

Sample Output 2

Tiger

Answer:(penalty regime: 0 %)

1. #include<stdio.h>
2. int main()
3. {
4. int year;
5. scanf("%d",&year);
6. if(year%12==8)
7. {
8. printf("Dragon");

9 }

1. else if(year%12==9)
2. {
3. printf("Snake");

13 }

1. else if(year%12==10)
2. {
3. printf("Horse"); 24



**Feedback**

**Input Expected Got**

2004 Monkey Monkey

2010 Tiger Tiger

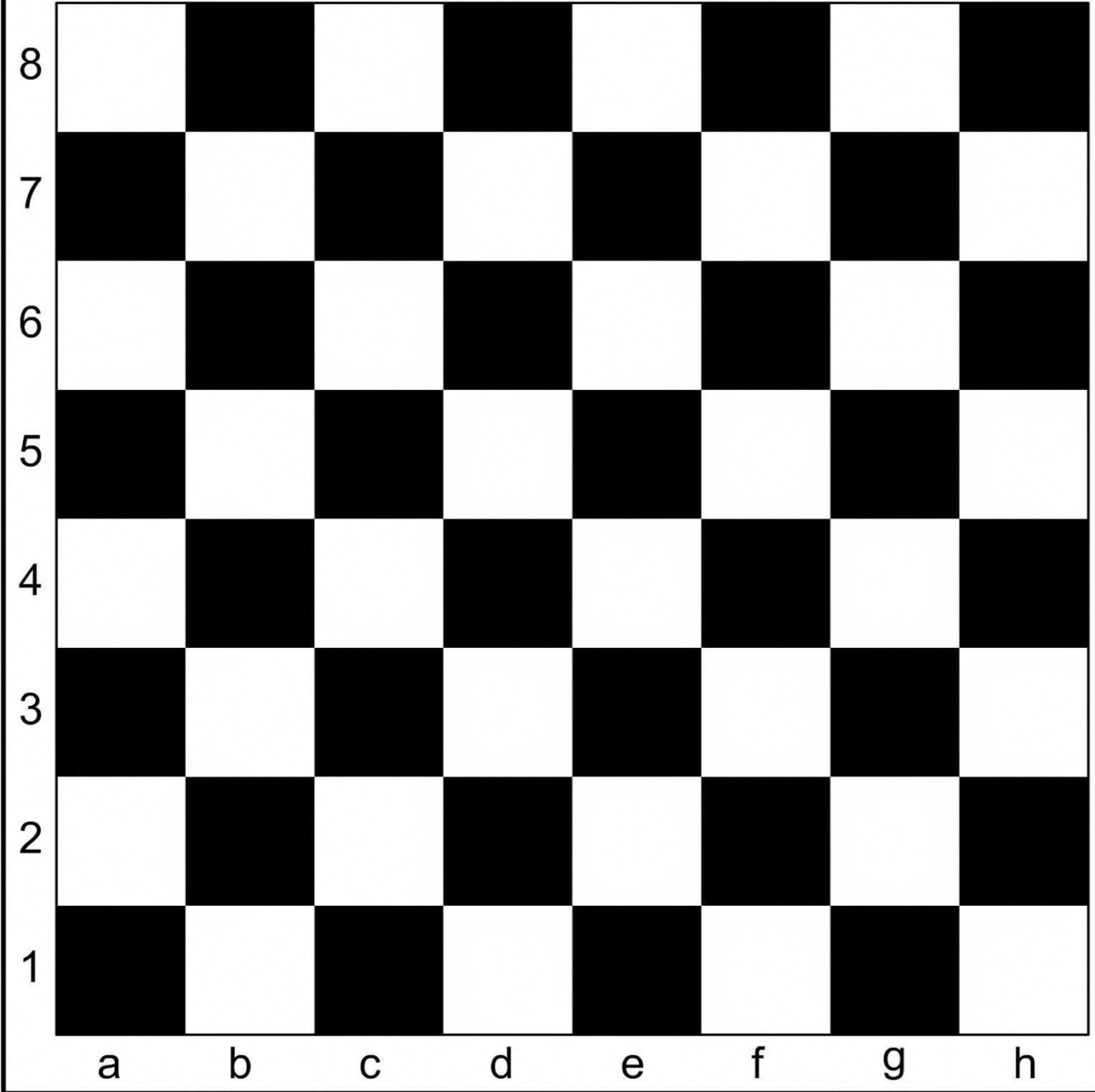
Passed all tests!

## **Question 3**

Correct

Marked out of 7.00

**Question text**

Positions on a chess board are identiﬁed by a letter and a number. The letter identiﬁes the column, while the number identiﬁes the row, as shown below:



25

Write a program that reads a position from the user. Use an if statement to determine if the column begins with a black square or a white square. Then use modular arithmetic to report the color of the square in that row. For example, if the user enters a1 then your program should report that the square is black. If the user enters d5 then your program should report that the square is white. Your program may assume that a valid position will always be entered. It does not need to perform any error checking.

Sample Input 1

a 1

Sample Output 1

The square is black.

Sample Input 2

d 5

Sample Output 2

The square is white.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | int main()  {  int num,sum; char alpha;  scanf("%c%d",&alpha,&num); sum=alpha+num; if(sum%2==0)  {  printf("The square is black.");  }  else  {  printf("The square is white.");  }  } |  |
|  | | | |

**Feedback**

**Input Expected Got**

a 1 The square is black. The square is black.

d 5 The square is white. The square is white.

Passed all tests!

26

27



REC-First Year-2023

# **GE23131-Programming using C-2023**

**Started on** Friday, 27 October 2023, 2:02 PM

**State** Finished

**Completed on** Friday, 27 October 2023, 2:50 PM

**Time taken** 48 mins 33 secs

## **Question 1**

Correct

Marked out of 3.00

**Question text**

Some data sets specify dates using the year and day of year rather than the year, month, and day of month. The day of year (DOY) is the sequential day number starting with day 1 on January 1st.

There are two calendars - one for normal years with 365 days, and one for leap years with 366 days. Leap years are divisible by 4. Centuries, like 1900, are not leap years unless they are divisible by 400. So, 2000 was a leap year.

To find the day of year number for a standard date, scan down the Jan column to find the day of month, then scan across to the appropriate month column and read the day of year number. Reverse the process to find the standard date for a given day of year.

Write a program to print the Day of Year of a given date, month and year.

Sample Input 1

18

6

2020

Sample Output 1

170

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 28 |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13 | int main()  {  int d,m,y,feb; scanf("%d%d%d",&d,&m,&y); if((y%100==0&&y%400)||(y%4==0))  feb=29; else feb=28; switch(m)  {  case 1: printf("%d",d); |  |

| 145 |  | bcraesaek;2: |  |  |
| --- | --- | --- | --- | --- |
| 16 | printf("%d",31+d); |
| 17 | break; |
| 18 | case 3: |
| 19 | printf("%d",31+feb+d); |
| 20 | break; |
| 21 | case 4: |
| 22 | printf("%d",31+feb+31+d); |
| 23 | break; |
| 24 | case 5: |
| 25 | printf("%d",31+feb+31+30+d); |
| 26 | break; |
| 27 | case 6: |
| 28 | printf("%d",31+feb+31+30+31+d); |
| 29 | break; |
| 30 | case 7: |
| 31 | printf("%d",31+feb+31+30+31+30+d); |
| 32 | break; |
| 33 | case 8: |
| 34 | printf("%d",31+feb+31+30+31+30+31+d); |
| 35 | break; |
| 36 | case 9: |
| 37 | printf("%d",31+feb+31+30+31+30+31+31+d); |
| 38 | break; |
| 39 | case 10: |
| 40 | printf("%d",31+feb+31+30+31+30+31+31+30+d); |
| 41 | break; |
| 42 | case 11: |
| 43 | printf("%d",31+feb+31+30+31+30+31+31+30+31+d); |
| 44 | break; |
| 45 | case 12: |
| 46 | printf("%d",31+feb+31+30+31+30+31+31+30+31+30+d); |
| 47 | break; |
| 48 |  | } |
| 49 | } |  |
|  | | | | |

**Feedback**

**Input Expected Got**

18

6

2020

170 170

Passed all tests!

## **Question 2**

Correct

Marked out of 5.00

**Question text**

Suppandi is trying to take part in the local village math quiz. In the first round, he is asked about shapes and areas. Suppandi, is confused, he was never any good at math. And also, he is bad at remembering the names of shapes.

Instead, you will be helping him [calculate the area](http://rajalakshmiinstitutions.net/moodle/mod/quiz/view.php?id=6) of shapes.

* When he says rectangle he is actually referring to a square.
* When he says square, he is actually referring to a triangle.
* When he says triangle he is referring to a rectangle
* And when he is confused, he just says something random. At this point, all you can do is say 0.

Help Suppandi by printing the correct answer in an integer.

29

Input Format

* Name of shape (always in upper case R à Rectangle, S à Square, T à Triangle)
* Length of 1 side
* Length of other side

Note: In case of triangle, you can consider the sides as height and length of base

Output Format

* Print the area of the shape.

Sample Input 1

T 10

20

Sample Output 1

200

Sample Input 2

S 30

40

Sample Output 2

600

Sample Input 3

R 10

10

Sample Output 3 30

100

Sample Input 4

G 8

8

Sample Output 4

0

Sample Input

C 9

10

Sample Output 4

0

Explanation:

* First is output of area of rectangle
* Then, output of area of triangle
* Then output of area square
* Finally, something random, so we print 0

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 31 |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | int main()  {  int a,b; char c;  scanf("%c%d%d",&c,&a,&b); switch(c)  {  case 'R': printf("%d",a\*b); break;  case 'S': printf("%.0f",(0.5)\*a\*b); break;  case 'T': printf("%d",a\*b); break;  default: printf("0"); |  |



| **2**01  22 |  | } |  |  |
| --- | --- | --- | --- | --- |
| } | |
|  | | | | |

**Feedback**

**Input Expected Got**

| T  10 | 200 | 200 |
| --- | --- | --- |
| 20 |  |  |
| S 30 | 600 | 600 |
| 40 |  |  |
| B 2 | 0 | 0 |
| 11 |  |  |
| R 10 | 300 | 300 |
| 30 |  |  |
| S 40 | 1000 | 1000 |
| 50 |  |  |

Passed all tests!

## **Question 3**

Correct

Marked out of 7.00

**Question text**

Superman is planning a journey to his home planet. It is very important for him to know which day he arrives there. They don't follow the 7-day week like us. Instead, they follow a 10-day week with the following days: Day Number Name of Day 1 Sunday 2 Monday 3 Tuesday 4 Wednesday 5 Thursday 6 Friday 7 Saturday 8 Kryptonday 9 Coluday 10 Daxamday Here are the rules of the calendar: • The calendar starts with Sunday always. • It has only 296 days. After the 296th day, it goes back to Sunday. You begin your journey on a Sunday and will reach after n. You have to tell on which day you will arrive when you reach there. Input format: • Contain a number n (0 < n) Output format: Print the name of the day you are arriving on Example Input 7 Example Output Kryptonday Example Input 1 Example Output Monday

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 32 |
| --- | --- | --- | --- |
| 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 | int main()  {  int n,day; scanf("%d",&n); if(n<296) day=n;  else  day=n-296; day%=10; day=day+1; day%=10; switch(day)  {  case 1: printf("Sunday"); break;  case 2: printf("Monday"); break;  case 3: printf("Tuesday"); break;  case 4: printf("Wednesday"); break; |  |

| 27 |  | case 5: |  |  |
| --- | --- | --- | --- | --- |
| 28 | printf("Thursday"); |
| 29 | break; |
| 30 | case 6: |
| 31 | printf("Friday"); |
| 32 | break; |
| 33 | case 7: |
| 34 | printf("Saturday"); |
| 35 | break; |
| 36 | case 8: |
| 37 | printf("Kryptonday"); |
| 38 | break; |
| 39 | case 9: |
| 40 | printf("Coluday"); |
| 41 | break; |
| 42 | case 10: |
| 43 | printf("Daxamday"); |
| 44 | break; |
| 45 |  |
| 46 |  | } |
| 47 | } |  |
|  | | | | |

**Feedback**

**Input Expected Got**

7 Kryptonday Kryptonday

1 Monday Monday

Passed all tests!

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Tuesday, 26 November 2024, 8:24 AM

**State** Finished

**Completed on** Tuesday, 26 November 2024, 8:37 AM

**Time taken** 12 mins 28 secs

## **Question 1**

Correct

Marked out of 3.00

**Question text**

Alice and Bob are playing a game called "Stone Game". Stone game is a two-player game. Let N be the total number of stones. In each turn, a player can remove either one stone or four stones. The player who picks the last stone, wins.

They follow the "Ladies First" norm. Hence Alice is always the one to make the first move. Your task is to find out whether Alice can win, if both play the game optimally.

Input Format

First line starts with T, which is the number of test cases. Each test case will contain N number of stones.

Output Format

Print "Yes" in the case Alice wins, else print "No".

Constraints

1<=T<=1000

1<=N<=10000

Sample Input and Output

Input

3

1

6

7

Output

Yes Yes No

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | int main()  {  int T,i=0,n,t; scanf("%d",&T); while(i<T)  {  scanf("%d",&n); t=n/4;  if(t%2==0 && n%2==0)  {  printf("No\n");  }  else if(t%2==1 && n%2==1)  {  printf("No\n");  }  else  {  printf("Yes\n");  } i++;  }  } |  |
|  | | | |

**Feedback**

**Input Expected Got**

3 Yes

1 Yes

6 No

7

Yes Yes No

Passed all tests!

## **Question 2**

Correct

Marked out of 5.00

**Question text**

You are designing a poster which prints out numbers with a unique style applied to each of them. The styling is based on the number of closed paths or holes present in a given number.

The number of holes that each of the digits from 0 to 9 have are equal to the number of closed paths in the digit. Their values are:

1, 2, 3, 5, and 7 = 0 holes.

0, 4, 6, and 9 = 1 hole.

8 = 2 holes.

Given a number, you must determine the sum of the number of holes for all of its digits. For example, the number 819 has 3 holes.

Complete the program, it must must return an integer denoting the total number of holes in num.

Constraints

1 ≤ num ≤ 109

Input Format For Custom Testing

There is one line of text containing a single integer num, the value to process.

Sample Input

630

Sample Output

2

Explanation

Add the holes count for each digit, 6, 3 and 0. Return 1 + 0 + 1 = 2.

Sample Case 1

Sample Input

1288

Sample Output

4

Explanation

Add the holes count for each digit, 1, 2, 8, 8. Return 0 + 0 + 2 + 2 = 4.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 36 |
| --- | --- | --- | --- |
| 2  3 | int main()  { |  |



| 4 | } | int a,b,n=0; scanf("%d",&a);  while(a>0)  {  b=a%10;  if(b==0 || b==6 || b==9 || b==4)  {  n=n+1;  }  else if(b==8)  {  n=n+2;  }  a=a/10;  }  printf("%d",n); |  |  |
| --- | --- | --- | --- | --- |
| 5 |
| 6 |
| 7 |
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |
| 19 |
| 20 |
| 21 |
| 22 |
|  | | | | |

**Feedback**

**Input Expected Got**

| 630 | 2 | 2 |
| --- | --- | --- |
| 1288 | 4 | 4 |

Passed all tests!

## **Question 3**

Correct

Marked out of 7.00

**Question text**

The problem solvers have found a new Island for [coding](http://rajalakshmiinstitutions.net/moodle/mod/quiz/view.php?id=38) and named it as Philaland. These smart people were given a task to make a purchase of items at the Island easier by distributing various coins with different values. Manish has come up with a solution that if we make coins category starting from $1 till the maximum price of the item present on Island, then we can purchase any item easily. He added the following example to prove his point.

Let’s suppose the maximum price of an item is 5$ then we can make coins of {$1, $2, $3, $4, $5}to purchase any item ranging from $1 till $5.

Now Manisha, being a keen observer suggested that we could actually minimize the number of coins required and gave following distribution {$1, $2, $3}. According to him any item can be purchased one time ranging from $1 to $5.

Everyone was impressed with both of them. Your task is to help Manisha come up with a minimum number of denominations for any arbitrary max price in Philaland.

**Input Format**

Contains an integer N denoting the maximum price of the item present on Philaland.

**Output Format**

Print a single line denoting the minimum number of denominations of coins required.

**Constraints** 37

1<=T<=100

1<=N<=5000

**Refer the sample output for formatting**

**Sample Input 1:**

10

**Sample Output 1:**

4

**Sample Input 2:**

5

**Sample Output 2:**

3

**Explanation:**

For test case 1, N=10.

According to Manish {$1, $2, $3,… $10} must be distributed.

But as per Manisha only {$1, $2, $3, $4} coins are enough to purchase any item ranging from $1 to $10. Hence minimum is 4. Likewise denominations could also be {$1, $2, $3, $5}. Hence answer is still 4.

For test case 2, N=5.

According to Manish {$1, $2, $3, $4, $5} must be distributed.

But as per Manisha only {$1, $2, $3} coins are enough to purchase any item ranging from $1 to $5. Hence minimum is

3. Likewise, denominations could also be {$1, $2, $4}. Hence answer is still 3.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 38 |
| --- | --- | --- | --- |



| 2  3  4  5  6  7  8  9  10  11  12 | int main() {  int n,r=0;  scanf("%d",&n); while(n!=0)  {  n=n/2; r=r+1;  }  printf("%d",r);  } |  |  |
| --- | --- | --- | --- |
|  | | | |

**Feedback**

**Input Expected Got**

| 10 | 4 | 4 |
| --- | --- | --- |
| 5 | 3 | 3 |
| 20 | 5 | 5 |
| 500 | 9 | 9 |
| 1000 | 10 | 10 |

Passed all tests!

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Tuesday, 26 December 2023, 8:39 AM

**State** Finished

**Completed on** Tuesday, 26 December 2023, 8:54 AM

**Time taken** 14 mins 27 secs

## **Question 1**

Correct

Marked out of 3.00

**Question text**

A set of N numbers (separated by one space) is passed as input to the program. The program must identify the count of numbers where the number is odd number.

Input Format:

The first line will contain the N numbers separated by one space.

Boundary Conditions:

3 <= N <= 50

The value of the numbers can be from -99999999 to 99999999

Output Format:

The count of numbers where the numbers are odd numbers.

Example Input / Output 1:

Input:

5 10 15 20 25 30 35 40 45 50

Output:

5

Explanation:

The numbers meeting the criteria are 5, 15, 25, 35, 45.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9 | int main() {  int n,x=0; while(scanf("%d",&n)==1) {  if(n%2!=0) {  x++; } }  printf("%d",x); return 0;  } |  |
|  | | | |

**Feedback**

**Input Expected Got**

5 10 15 20 25 30 35 40 45 50 5 5

Passed all tests!

## **Question 2**

Correct

Marked out of 5.00

**Question text**

Given a number N, return true if and only if it is a *confusing number*, which satisfies the following condition:

We can rotate digits by 180 degrees to form new digits. When 0, 1, 6, 8, 9 are rotated 180 degrees, they become 0, 1, 9,

8, 6 respectively. When 2, 3, 4, 5 and 7 are rotated 180 degrees, they become invalid. A *confusing number* is a number that when rotated 180 degrees becomes a **different** number with each digit valid.

**Example 1:**

6 -> 9

Input: 6 Output: true Explanation:

We get 9 after rotating 6, 9 is a valid number and 9!=6.

**Example 2:**

89 -> 68

Input: 89 41

Output: true Explanation:

We get 68 after rotating 89, 86 is a valid number and 86!=89.

**Example 3:**

11 -> 11

Input: 11 Output: false Explanation:

We get 11 after rotating 11, 11 is a valid number but the value remains the same, thus 11 is not a confusing number.

**Note:**

1. 0 <= N <= 10^9
2. After the rotation we can ignore leading zeros, for example if after rotation we have 0008 then this number is considered as just 8.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | int main() { |  |
| 3 | int n,x,y=1; |
| 4 | scanf("%d",&n); |
| 5 | while(n!=0 && y==1) { |
| 6 | x=n%10; n=n/10; |
| 7 | if(x==2 || x==3 || x==4 || x==7) { |
| 8 | y++; } } |
| 9 | if(y==1) { |
| 10 | printf("true"); } |
| 11 | else { |
| 12 | printf("false"); } |
| 13 |  |
| 14 | } |
|  | | | |

**Feedback**

**Input Expected Got**

| 6 | true | true |
| --- | --- | --- |
| 89 | true | true |
| 25 | false | false |

Passed all tests!

## **Question 3**

Correct

Marked out of 7.00

**Question text**

A nutritionist is labeling all the best power foods in the market. Every food item arranged in a single line, will have a value beginning from 1 and increasing by 1 for each, until all items have a value associated with them. An item's value is the same as the number of macronutrients it has. For example, food item with value 1 has 1 macronutrient, food item with value 2 has 2 macronutrients, and incrementing in this fashion.

The nutritionist has to recommend the best combination to patients, i.e. maximum total of macronutrients. However, the nutritionist must avoid prescribing a particular sum of macronutrients (an 'unhealthy' number), and this sum is known. The nutritionist chooses food items in the increasing order of their value. Compute the highest total of macronutrients that can be prescribed to a patient, without the sum matching the given 'unhealthy' number.

Here's an illustration:

Given *4* food items (hence value: *1,2,3* and *4*), and the unhealthy sum being *6* macronutrients, on choosing items *1, 2, 3*

-> the sum is *6,* which matches the 'unhealthy' sum. Hence, one of the three needs to be skipped. Thus, the best combination is from among:

· *2 + 3 + 4 = 9*

· *1 + 3 + 4 = 8*

· *1 + 2 + 4 = 7*

Since *2 + 3 + 4 = 9,* allows for maximum number of macronutrients, 9 is the right answer.

Complete the code in the editor below. It must return an integer that represents the maximum total of macronutrients, modulo *1000000007 (109 + 7)*.

It has the following:

*n:* an integer that denotes the number of food items

*k:* an integer that denotes the unhealthy number

**Constraints**

· *1 ≤ n ≤ 2 × 109*

· *1 ≤ k ≤ 4 × 1015*

Input Format For Custom Testing

The first line contains an integer, *n*, that denotes the number of food items. The second line contains an integer, *k*, that denotes the unhealthy number.

**Sample Input 0**

2

2

**Sample Output 0**

3

**Explanation 0**

The following sequence of *n = 2* food items:

1. Item 1 has 1 macronutrients.
2. *1 + 2 = 3*; observe that this is the max total, and having avoided having exactly *k = 2* macronutrients.

**Sample Input 1**

2

1

**Sample Output 1**

2

**Explanation 1**

1. Cannot use item *1* because *k = 1* and *sum ≡ k* has to be avoided at any time.
2. Hence, max total is achieved by *sum = 0 + 2 = 2*.

Sample Case 2

**Sample Input For Custom Testing**

**Sample Input 2**

3

3

**Sample Output 2**

5

**Explanation 2**

*2 + 3 = 5, i*s the best case for maximum nutrients.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10 | int main() {  long long int n,t,i,nut=0; scanf("%lld %lld",&n,&t); for(i=1;i<=n;i++) {  nut=nut+i; if(nut==t) {  nut=nut-1; } } printf("%lld",nut%1000000007);  } |  |
|  | | | |

**Feedback**

**Input Expected Got**

2 3 3

2

2 2 2

1

3 5 5

3

Passed all tests!

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Friday, 31 December 2024, 1:28 PM

**State** Finished

**Completed on** Friday, 31 December 2024, 1:55 PM

**Time taken** 26 mins 43 secs

## **Question 1**

Correct

Marked out of 3.00

**Question text**

Write a program that prints a simple chessboard.

Input format:

The first line contains the number of inputs T.

The lines after that contain a different values for size of the chessboard

Output format:

Print a chessboard of dimensions size \* size. Print a Print W for white spaces and B for black spaces.

Input:

2

3

5

Output:

WBW BWB WBW WBWBW BWBWB WBWBW BWBWB WBWBW

| 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  32  33  34 | #include<stdio.h>  int main()  {  int T,d,i=0,i1,i2,o; char c; scanf("%d",&T); while(i<T)  {  scanf("%d",&d); i1=0;  while(i1<d)  {  o=1; i2=0;  if(i1%2==0)  {  o=0;  }  while(i2<d)  {  c='B';  if(i2%2==o)  {  c='W';  }  printf("%c",c); i2++;  } i1+=1;  printf("\n");  }  i=i+1;  }  } |  |  |
| --- | --- | --- | --- |
|  | | | |

**Feedback**

**Input Expected Got**

WBW BWB WBW

2 WBWBW

3 BWBWB

5 WBWBW

BWBWB WBWBW

WBW BWB WBW WBWBW BWBWB WBWBW BWBWB WBWBW



Passed all tests!

## **Question 2**

Correct

Marked out of 5.00

**Question text**

Let’s print a chessboard!

Write a program that takes input:

The first line contains T, the number of test cases

Each test case contains an integer N and also the starting character of the chessboard

Output Format

Print the chessboard as per the given examples

Sample Input / Output

Input:

2

2 W

3 B

Output:

WB BW BWB WBW BWB

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | int main() |  |
| 3 | { |
| 4 | int T,d,i,i1,i2,o,z; |
| 5 | char c,s; |
| 6 | scanf("%d",&T); |
| 7 | for(i=0;i<T;i++) |
| 8 | { |
| 9 | scanf("%d %c",&d,&s); |
| 10 | for(i1=0;i1<d;i1++) |
| 11 | { |
| 12 | z=(s=='W') ? 0:1; |
| 13 | o=(i1%2==z) ? 0:1; |
| 14 | for(i2=0;i2<d;i2++) |
| 15 | { |
| 16 | c=(i2%2==o) ? 'W' : 'B'; |
| 17 | printf("%c",c); |
| 18 | } |
| 19 | printf("\n"); |
| 20 | } |
| 21 |  |
| 22 | } |
| 23 | return 0; |
| 24 | } |
|  | | | |

**Feedback**

**Input Expected Got**

|  | WB | WB |
| --- | --- | --- |
| 2 | BW | BW |
| 2 W | BWB | BWB |
| 3 B | WBW | WBW |
|  | BWB | BWB |

Passed all tests!

## **Question 3**

Correct 48

Marked out of 7.00

**Question text**

Decode the logic and print the Pattern that corresponds to given input.

If N= 3

then pattern will be :

10203010011012

\*\*4050809

\*\*\*\*607

If N= 4, then pattern will be:

1020304017018019020

\*\*50607014015016

\*\*\*\*809012013

\*\*\*\*\*\*10011

Constraints

2 <= N <= 100

Input Format

First line contains T, the number of test cases Each test case contains a single integer N

Output

First line print Case #i where i is the test case number In the subsequent line, print the pattern

Test Case 1

3

3

4

5

49

Output

Case #1

10203010011012

\*\*4050809

\*\*\*\*607

Case #2

1020304017018019020

\*\*50607014015016

\*\*\*\*809012013

\*\*\*\*\*\*10011

Case #3

102030405026027028029030

\*\*6070809022023024025

\*\*\*\*10011012019020021

\*\*\*\*\*\*13014017018

\*\*\*\*\*\*\*\*15016

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 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 | int main(){  int n,v,p3,c,in,i,i1,i2,t,ti; scanf("%d",&t); for(ti=0;ti<t;ti++){  v=0;  scanf("%d",&n); printf("Case #%d\n",ti+1); for(i=0;i<n;i++){  c=0;  if(i>0){  for(i1=0;i1<i;i1++) printf("\*\*");  }  for(i1=i;i1<n;i1++){ if(i>0) c++; printf("%d0",++v);  }  if(i==0){  p3=v+(v\*(v-1))+1;  in=p3;  }  in=in-c; p3=in;  for(i2=i;i2<n;i2++){ printf("%d",p3++); if(i2!=n-1) printf("0");  }printf("\n");  }  }  } |  |
|  | | | |

**Feedback**

**Input Expected Got**

Case #1

10203010011012

\*\*4050809

\*\*\*\*607

Case #2

1020304017018019020

Case #1

10203010011012

\*\*4050809

\*\*\*\*607

Case #2

1020304017018019020 50

| 3 | \*\*50607014015016 | \*\*50607014015016 |
| --- | --- | --- |
| 3 | \*\*\*\*809012013 | \*\*\*\*809012013 |
| 4 | \*\*\*\*\*\*10011 | \*\*\*\*\*\*10011 |
| 5 | Case #3 | Case #3 |

102030405026027028029030 102030405026027028029030

\*\*6070809022023024025

\*\*\*\*10011012019020021

\*\*\*\*\*\*13014017018

\*\*\*\*\*\*\*\*15016

\*\*6070809022023024025

\*\*\*\*10011012019020021

\*\*\*\*\*\*13014017018

\*\*\*\*\*\*\*\*15016

Passed all tests!

51

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Tuesday, 31 December 2024, 9:15 AM

**State** Finished

**Completed on** Tuesday, 31 December 2026, 9:28 AM

**Time taken** 12 mins 56 secs

## **Question 1**

Correct

Marked out of 3.00

**Question text**

The k-digit number N is an Armstrong number if and only if the k-th power of each digit sums to N.

Given a positive integer N, return true if and only if it is an Armstrong number.

Example 1:

Input:

153

Output:

true

Explanation:

153 is a 3-digit number, and 153 = 1^3 + 5^3 + 3^3.

Example 2:

Input:

123

Output:

false

Explanation: 52

123 is a 3-digit number, and 123 != 1^3 + 2^3 + 3^3 = 36.

Example 3:

Input:

1634

Output:

true

Note:

1 <= N <= 10^8

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | #include<math.h> |  |
| 3 | int main() |
| 4 | { |
| 5 | int n; |
| 6 | scanf("%d",&n); |
| 7 | int x=0,n2=n; |
| 8 | while(n2!=0) |
| 9 | { |
| 10 | x++; |
| 11 | n2=n2/10; |
| 12 |  |
| 13 | } |
| 14 | int sum=0; |
| 15 | int n3=n,n4; |
| 16 | while(n3!=0) |
| 17 | { |
| 18 | n4=n3%10; |
| 19 | sum = sum+pow(n4,x); |
| 20 | n3=n3/10; |
| 21 |  |
| 22 | } |
| 23 | if(n==sum) |
| 24 | { |
| 25 | printf("true"); |
| 26 | } |
| 27 | else |
| 28 | { |
| 29 | printf("false"); |
| 30 |  |
| 31 | } |
| 32 | return 0; |
| 33 | } |
|  | | | |

**Feedback**

**Input Expected Got**

153 true true

123 false false 53

Passed all tests!

## **Question 2**

Correct

Marked out of 5.00

**Question text**

Take a number, reverse it and add it to the original number until the obtained number is a palindrome. Constraints 1<=num<=99999999 Sample Input 1 32 Sample Output 1 55 Sample Input 2 789 Sample Output 2 66066 Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | int main() |  |
| 3 | { |
| 4 | int rn, n,nt=0,i=0; |
| 5 | scanf("%d",&n); |
| 6 | do{ |
| 7 | nt=n;rn=0; |
| 8 | while(n!=0) |
| 9 | { |
| 10 | rn=rn\*10 + n%10; |
| 11 | n=n/10; |
| 12 |  |
| 13 | } |
| 14 | n=nt+rn; |
| 15 | i++; |
| 16 |  |
| 17 | } |
| 18 | while(rn!=nt || i==1); |
| 19 | printf("%d",rn); |
| 20 | return 0; |
| 21 | } |
|  | | | |

**Feedback**

**Input Expected Got**

| 32 | 55 | 55 |
| --- | --- | --- |
| 789 | 66066 | 66066 |

Passed all tests!

## **Question 3**

Correct

Marked out of 7.00

**Question text**

A number is considered lucky if it contains either 3 or 4 or 3 and 4 both in it. Write a program to print the nth lucky number. Example, 1st lucky number is 3, and 2nd lucky number is 4 and 3rd lucky number is 33 and 4th lucky number is 34 and so on. Note that 13, 40 etc., are not lucky as they have other numbers in it.

The program should accept a number 'n' as input and display the nth lucky number as output.

Sample Input 1:

3

Sample Output 1: 54

33

Explanation:

Here the lucky numbers are 3, 4, 33, 34., and the 3rd lucky number is 33.

Sample Input 2:

34

Sample Output 2:

33344

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | int main() |  |
| 3 | { |
| 4 | int n=1,i=0,nt,co=0,e; |
| 5 | scanf("%d",&e); |
| 6 | while(i<e) |
| 7 | { |
| 8 | nt=n; |
| 9 | while(nt!=0) |
| 10 | { |
| 11 | co=0; |
| 12 | if(nt%10!=3 && nt%10!=4) |
| 13 | { |
| 14 | co=1; |
| 15 | break; |
| 16 |  |
| 17 | } |
| 18 | nt=nt/10; |
| 19 | } |
| 20 | if(co==0) |
| 21 | { |
| 22 | i++; |
| 23 | } |
| 24 | n++; |
| 25 | } |
| 26 | printf("%d",--n); |
| 27 | return 0; |
| 28 | } |
|  | | | |

**Feedback**

**Input Expected Got**

34 33344 33344

Passed all tests!

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Friday, 29 December 2024, 1:57 PM

**State** Finished

**Completed on** Friday, 29 December 2024, 2:15 PM

**Time taken** 18 mins 5 secs

## **Question 1**

Correct

Marked out of 3.00 Flag question

**Question text**

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[i] - A[j] = k, i != j.

Input Format

1. First line is number of test cases T. Following T lines contain:
2. N, followed by N integers of the array
3. The non-negative integer k

Output format

Print 1 if such a pair exists and 0 if it doesn’t.

Example

Input:

1

3 1 3 5

4

Output:

1

Input:

1

3 1 3 5

99

Output:

0

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | int main(){  int t; scanf("%d",&t);  while(t--){  int n; scanf("%d",&n); int a[n];  for(int i=0;i<n;i++){ scanf("%d",&a[i]);  }  int k; scanf("%d",&k); int flag=0;  for(int i=0;i<n;i++){ for(int j=i+1;j<n;j++){  if(a[i]-a[j]==k || a[j]-a[i]==k){flag=1;break;}  }  if(flag) break;} printf("%d\n",flag);  }  } |  |
|  | | | |

**Feedback**

**Input Expected Got**

| 1 |  |
| --- | --- |
| 3 1 3 5 1 | 1 |
| 4 |  |
| 1 |  |
| 3 1 3 5 0 | 0 |
| 99 |  |

Passed all tests!

## **Question 2**

Correct

Marked out of 5.00

**Question text**

Sam loves chocolates and starts buying them on the 1st day of the year. Each day of the year, x, is numbered from 1 to

Y. On days when x is odd, Sam will buy x chocolates; on days when x is even, Sam will not purchase any chocolates.

Complete the code in the editor so that for each day Ni (where 1 ≤ x ≤ N ≤ Y) in array arr, the number of chocolates Sam purchased (during days 1 through N) is printed on a new line. This is a function-only challenge, so input is handled for you by the locked stub code in the editor.

Input Format

The program takes an array of integers as a parameter.

The locked code in the editor handles reading the following input from stdin, assembling it into an array of integers (arr), and calling calculate(arr).

The first line of input contains an integer, T (the number of test cases). Each line i of the T subsequent lines describes the ith test case as an integer, Ni (the number of days).

Constraints

1 ≤ T ≤ 2 × 105

1 ≤ N ≤ 2 × 106

1 ≤ x ≤ N ≤ Y

Output Format

For each test case, Ti in arr, your calculate method should print the total number of chocolates Sam purchased by day Ni on a new line.

Sample Input 0

3

1

2

3

Sample Output 0

1

1

4

Explanation

Test Case 0: N = 1

Sam buys 1 chocolate on day 1, giving us a total of 1 chocolate. Thus, we print 1 on a new line.

Test Case 1: N = 2

Sam buys 1 chocolate on day 1 and 0 on day 2. This gives us a total of 1 chocolate. Thus, we print 1 on a new line.

Test Case 2: N = 3

Sam buys 1 chocolate on day 1, 0 on day 2, and 3 on day 3. This gives us a total of 4 chocolates. Thus, we print 4 on a new line. 58

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | int main(){ |  |
| 3 | int t; |
| 4 | scanf("%d",&t); |
| 5 | while(t--){ |
| 6 | int n,c=0; |
| 7 | scanf("%d",&n); |
| 8 | for(int i=0;i<=n;i++){ |
| 9 | if(i%2!=0) c=c+i; |
| 10 | }printf("%d\n",c); |
| 11 | } |
| 12 |  |
| 13 | } |
|  | | | |

**Feedback**

**Input Expected Got**

3 1 1

1 1 1

2 4 4

3

| 1296 | 1296 |
| --- | --- |
| 2500 | 2500 |
| 1849 | 1849 |
| 729 | 729 |
| 400 | 400 |
| 25 | 25 |
| 1521 | 1521 |
| 25 | 25 |
| 49 | 49 |
| 2401 | 2401 |

10

71

100

86

54

40

9

77

9

13

98

Passed all tests!

## **Question 3**

Correct

Marked out of 7.00 

**Question text**

The number of goals achieved by two football teams in matches in a league is given in the form of two lists. Consider:

* Football team A, has played three matches, and has scored { 1 , 2 , 3 } goals in each match respectively.
* Football team B, has played two matches, and has scored { 2, 4 } goals in each match respectively.
* Your task is to compute, for each match of team B, the total number of matches of team A, where team A has scored less than or equal to the number of goals scored by team B in that match.
* In the above case:
* For 2 goals scored by team B in its first match, team A has 2 matches with scores 1 and 2.
* For 4 goals scored by team B in its second match, team A has 3 matches with scores 1, 2 and 3.

Hence, the answer: {2, 3}.

Complete the code in the editor below. The program must return an array of m positive integers, one for each maxes[i] representing the total number of elements nums[j] satisfying nums[j] ≤ maxes[i] where 0 ≤ j < n and 0 ≤ i < m, in the given order.

It has the following:

nums[nums[0],...nums[n-1]]: first array of positive integers maxes[maxes[0],...maxes[n-1]]: second array of positive integers

Constraints

* 2 ≤ n, m ≤ 105
* 1 ≤ nums[j] ≤ 109, where 0 ≤ j < n.
* 1 ≤ maxes[i] ≤ 109, where 0 ≤ i < m.

Input Format For Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer n, the number of elements in nums.

The next n lines each contain an integer describing nums[j] where 0 ≤ j < n. The next line contains an integer m, the number of elements in maxes.

The next m lines each contain an integer describing maxes[i] where 0 ≤ i < m.

Sample Case 0

Sample Input 0

4

1

4

2

4

2

3

5

Sample Output 0

4

Explanation 0

We are given n = 4, nums = [1, 4, 2, 4], m = 2, and maxes = [3, 5].

1. For maxes[0] = 3, we have 2 elements in nums (nums[0] = 1 and nums[2] = 2) that are ≤ maxes[0].
2. For maxes[1] = 5, we have 4 elements in nums (nums[0] = 1, nums[1] = 4, nums[2] = 2, and nums[3] = 4) that are

≤ maxes[1].

Thus, the function returns the array [2, 4] as the answer.

Sample Case 1

Sample Input 1

5

2

10

5

4

8

4

3

1

7

8

Sample Output 1

1

0

3

4

Explanation 1

We are given, n = 5, nums = [2, 10, 5, 4, 8], m = 4, and maxes = [3, 1, 7, 8].

1. For maxes[0] = 3, we have 1 element in nums (nums[0] = 2) that is ≤ maxes[0].
2. For maxes[1] = 1, there are 0 elements in nums that are ≤ maxes[1].
3. For maxes[2] = 7, we have 3 elements in nums (nums[0] = 2, nums[2] = 5, and nums[3] = 4) that are ≤ maxes[2].
4. For maxes[3] = 8, we have 4 elements in nums (nums[0] = 2, nums[2] = 5, nums[3] = 4, and nums[4] = 8) that are

Thus, the function returns the array [1, 0, 3, 4] as the answer.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | int main()  {  int s1,s2,ans; scanf("%d",&s1); int ta[s1];  for(int i=0;i<s1;i++) scanf("%d",&ta[i]);  scanf("%d",&s2); int tb[s2];  for(int i=0;i<s2;i++) scanf("%d",&tb[i]); for(int j=0;j<s2;j++)  {  ans=0;  for(int i=0;i<s1;i++){ if(tb[j]>=ta[i]) ans++;  }printf("%d\n",ans);  }  } |  |
|  | | | |

**Feedback**

**Input Expected Got**

4

1

| 2 | 2 |
| --- | --- |
| 4 | 4 |
| 1 | 1 |
| 0 | 0 |
| 3 | 3 |
| 4 | 4 |

4

2

4

2

3

5

5

2

10

5

4

8

4

3

1

7

8

Passed all tests!

REC-First Year-2024

# **GE23131-Programming using C-2023**

**Started on** Monday, 8 January 2024, 8:42 AM

**State** Finished

**Completed on** Monday, 8 January 2024, 9:00 AM

**Time taken** 18 mins 7 secs

## **Question 1**

Correct

Marked out of 1.00

**Question text**

Given an array of numbers and a window of size k. Print the maximum of numbers inside the window for each step as the window moves from the beginning of the array.

Input Format

Input contains the array size, no of elements and the window size Output Format

Print the maximum of numbers Constraints

1 <= size <= 1000

Sample Input 1

8

1 3 5 2 1 8 6 9

3

Sample Output 1

5 5 5 8 8 9

For example:

**Input Result**

8

1 3 5 2 1 8 6 9

3

5 5 5 8 8 9



10

3 7 5 1 2 9 8 5 3 2 7 7 5 9 9 9 8 5

3

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 63 |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13  14 | int main()  {  int n,k; scanf("%d",&n); int arr[n];  for(int i=0;i<n;i++)  {  scanf("%d",&arr[i]);  }  scanf("%d",&k);  for(int a=0;a<=n-k;a++)  {  int max=arr[a]; |  |

| 15  16  17  18  19  20  21  22  23  24 |  | for(int b=a;b<a+k;b++)  {  if(arr[b]>max)  {  max=arr[b];  }  }  printf("%d ",max); |  |  |
| --- | --- | --- | --- | --- |
| }  } | |
|  | | | | |

**Feedback**

**Input Expected Got**

8

1 3 5 2 1 8 6 9

3

5 5 5 8 8 9 5 5 5 8 8 9



10

3 7 5 1 2 9 8 5 3 2 7 7 5 9 9 9 8 5 7 7 5 9 9 9 8 5

3

Passed all tests!

## **Question 2**

Correct

Marked out of 1.00

**Question text**

Given an array and a threshold value find the output. Input: {5,8,10,13,6,2}

Threshold = 3 Output count = 17 Explanation:

Number Parts Counts

| 5 | {3,2} | 2 |  |
| --- | --- | --- | --- |
| 8 | {3,3,2} |  | 3 |
| 10 | {3,3,3,1} |  | 4 |
| 13 | {3,3,3,3,1} |  | 5 |
| 6 | {3,3} |  | 2 |
| 2 | {2} |  | 1 |

Input Format

N - no of elements in an array Array of elements

Threshold value Output Format Display the count Sample Input 1

6

5 8 10 13 6 2

64

3

Sample Output 1

17

For example:

**Input Result**

6

5 8 10 13 6 2 17

3

7

20 35 57 30 56 87 30 33

10

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | int main()  {  int n,t,count=0; scanf("%d",&n); int arr[n];  for(int i=0;i<n;i++)  {  scanf("%d",&arr[i]);  }  scanf("%d",&t); for(int j=0;j<n;j++)  {  while(arr[j]>0)  {  arr[j]-=t; count++;  }  }  printf("%d",count);  } |  |
|  | | | |

**Feedback**

**Input Expected Got**

| 6 |  | |
| --- | --- | --- |
| 5 8 10 13 6 2 | 17 | 17 |
| 3 |  |  |
| 7 |  |  |
| 20 35 57 30 56 87 | 30 33 | 33 |
| 10 |  |  |

Passed all tests!

## **Question 3**

Correct

Marked out of 1.00

**Question text**

Output is a merged array without duplicates. Input Format

N1 - no of elements in array 1 Array elements for array 1

Array elements for array2 Output Format

Display the merged array Sample Input 1

5

1 2 3 6 9

4

2 4 5 10

Sample Output 1

1 2 3 4 5 6 9 10

For example:

**Input Result**

5

1 2 3 6 9 1 2 3 4 5 6 9 10

4

2 4 5 10

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 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  32  33  34  35  36  37  38  39  40  41 | int main()  {  int a,b; scanf("%d",&a); int arr1[a];  for(int i=0;i<a;i++) scanf("%d",&arr1[i]);  scanf("%d",&b); int arr2[b];  for(int i=0;i<b;i++) scanf("%d",&arr2[i]); int p=0,q=0; while((p<a)&&(q<b))  {  if(arr1[p]<arr2[q])  {  printf("%d ",arr1[p]); p++;  }  else if(arr1[p]>arr2[q])  {  printf("%d ",arr2[q]); q++;  }  else  {  printf("%d ",arr1[p]); p++;  q++;  }  }  for(int j=p;j<a;j++)  {  printf("%d ",arr1[j]);  }  for(int j=q;j<b;j++)  {  printf("%d ",arr2[j]);  }  } |  |
|  | | | |

**Input Expected Got**

5

1 2 3 6 9 1 2 3 4 5 6 9 10 1 2 3 4 5 6 9 10

4

2 4 5 10

Passed all tests!

REC-First Year-2024

# **GE23131-Programming using C-2023**

**Started on** Friday, 29 December 2024, 2:19 PM

**State** Finished

**Completed on** Friday, 29 December 2024, 2:50 PM

**Time taken** 31 mins 8 secs

## **Question 1**

Correct

Marked out of 1.00

**Question text**

Sunny and Johnny like to pool their money and go to the ice cream parlor. Johnny never buys the same flavor that Sunny does. The only other rule they have is that they spend all of their money.

Given a list of prices for the flavors of ice cream, select the two that will cost all of the money they have.

For example, they have ***m = 6*** to spend and there are flavors costing ***cost = [1, 2, 3, 4, 5, 6]***. The two flavors costing

***1*** and ***5*** meet the criteria. Using ***1***-based indexing, they are at indices ***1*** and ***4***.

**Function Description**

Complete the code in the editor below. It should return an array containing the indices of the prices of the two flavors they buy.

It has the following:

* m: an integer denoting the amount of money they have to spend
* cost: an integer array denoting the cost of each flavor of ice cream

**Input Format**

The first line contains an integer, ***t***, denoting the number of trips to the ice cream parlor. The next ***t*** sets of lines each describe a visit. Each trip is described as follows:

1. The integer ***m***, the amount of money they have pooled.
2. The integer ***n***, the number of flavors offered at the time.
3. ***n*** space-separated integers denoting the cost of each flavor: ***cost[cost[1], cost[2], . . . , cost[n]]***.

**Note:** The index within the cost array represents the flavor of the ice cream purchased.

**Constraints**

· ***1 ≤ t ≤ 50***

· ***2 ≤ m ≤ 104***

· ***2 ≤ n ≤ 104***

· ***1 ≤ cost[i] ≤ 104, " i Î [1, n]***

· There will always be a unique solution.

**Output Format**

For each test case, print two space-separated integers denoting the indices of the two flavors purchased, in ascending order.

**Sample Input**

2

4

5

1 4 5 3 2

4

4

2 2 4 3

**Sample Output**

1 4

1 2

**Explanation**

Sunny and Johnny make the following two trips to the parlor:

1. The first time, they pool together ***m = 4*** dollars. Of the five flavors available that day, flavors ***1*** and ***4*** have a total cost of ***1 + 3 = 4***.
2. The second time, they pool together ***m = 4*** dollars. TOf the four flavors available that day, flavors ***1*** and ***2*** have a total cost of ***2 + 2 = 4***.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 69 |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11 | int main(){  int t,m,n,c=0; scanf("%d",&t); for(int i=0;i<t;i++){  c=0;  scanf("%d\n%d",&m,&n); int arr[n];  for(int j=0;j<n;j++){ scanf("%d",&arr[j]);  } |  |



| 12  13  14  15  16  17  18  19  20  21 |  | for(int a=0;a<n-1;a++){ for(int b=a+1;b<n;b++){  if(arr[a]+arr[b]==m){ printf("%d %d\n",a+1,b+1); c=1;break;  }  } if(c==1) break;  } |  |  |
| --- | --- | --- | --- | --- |
| }  return 0;} | |
|  | | | | |

**Feedback**

**Input Expected Got**

2

4

5 1 4 1 4

1 4 5 3 2 1 2 1 2

4

4

2 2 4 3

Passed all tests!

## **Question 2**

Correct

Marked out of 1.00

**Question text**

Numeros the Artist had two lists that were permutations of one another. He was very proud. Unfortunately, while transporting them from one exhibition to another, some numbers were lost out of the first list. Can you find the missing numbers?

As an example, the array with some numbers missing, ***arr = [7, 2, 5, 3, 5, 3]***. The original array of numbers ***brr = [7, 2, 5, 4, 6, 3, 5, 3]***. The numbers missing are ***[4, 6]***.

**Notes**

* If a number occurs multiple times in the lists, you must ensure that the frequency of that number in both lists is the same. If that is not the case, then it is also a missing number.
* You have to print all the missing numbers in ascending order.
* Print each missing number once, even if it is missing multiple times.
* The difference between maximum and minimum number in the second list is less than or equal to ***100***.

Complete the code in the editor below. It should return an array of missing numbers.

It has the following:

* arr: the array with missing numbers
* brr: the original array of numbers

**Input Format**

70

There will be four lines of input:

***n*** - the size of the first list, ***arr***

The next line contains ***n*** space-separated integers ***arr[i] m*** - the size of the second list, ***brr***

The next line contains ***m*** space-separated integers ***brr[i]***

**Constraints**

* ***1 ≤ n, m ≤ 2 x 105***
* ***n ≤ m***
* ***1 ≤ brr[i] ≤ 2 x 104***
* ***Xmax – Xmin < 101***

**Output Format**

Output the missing numbers in ascending order.

**Sample Input**

10

203 204 205 206 207 208 203 204 205 206

13

203 204 204 205 206 207 205 208 203 206 205 206 204

**Sample Output**

204 205 206

**Explanation**

***204*** is present in both arrays. Its frequency in ***arr*** is ***2***, while its frequency in ***brr*** is ***3***. Similarly, ***205*** and ***206*** occur twice in ***arr***, but three times in ***brr***. The rest of the numbers have the same frequencies in both lists.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 71 |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9 | int main(){  int n,m,c,c1=0,co; scanf("%d",&n); int arr[n];  for(int a=0;a<n;a++){ scanf("%d",&arr[a]);  }  scanf("%d",&m); |  |

| 10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44 | int brr[m],ans[m]; for(int b=0;b<m;b++){  scanf("%d",&brr[b]);  }  for(int j=0;j<m;j++)  {  c=0;  for(int i=0;i<n;i++){ if(arr[i]==brr[j]){  c=1;  arr[i]=-1; break;  }  }  if(c==0){  ans[c1]=brr[j]; c1++;  }  }  for(int a=0;a<c1;a++){ co=0;  for(int b=0;b<c1;b++){ if(ans[b]<ans[a]) co++;  }  int temp=ans[a]; ans[a]=ans[co]; ans[co]=temp;  }  for(int i=0;i<c1;i++) printf("%d ",ans[i]);  return 0;  } |  |  |
| --- | --- | --- | --- |
|  | | | |

**Feedback**

**Input Expected Got**

10

203 204 205 206 207 208 203 204 205 206

13

203 204 204 205 206 207 205 208 203 206 205 206 204

204 205 206 204 205 206



Passed all tests!

## **Question 3**

Correct

Marked out of 1.00

**Question text**

Watson gives Sherlock an array of integers. His challenge is to find an element of the array such that the sum of all elements to the left is equal to the sum of all elements to the right. For instance, given the array ***arr = [5, 6, 8, 11]***, ***8*** is between two subarrays that sum to ***11***. If your starting array is ***[1]***, that element satisfies the rule as left and right sum to ***0***.

You will be given arrays of integers and must determine whether there is an element that meets the criterion.

Complete the code in the editor below. It should return a string, either YES if there is an element meeting the criterion or NO otherwise.

It has the following:

* arr: an array of integers 72

**Input Format**

The first line contains ***T***, the number of test cases. The next ***T*** pairs of lines each represent a test case.

* The first line contains ***n***, the number of elements in the array ***arr***.
* The second line contains ***n*** space-separated integers ***arr[i]*** where ***0 ≤ i < n***.

**Constraints**

· ***1 ≤ T ≤ 10***

· ***1 ≤ n ≤ 105***

· ***1 ≤ arr[i] ≤ 2 x 104***

· ***0 ≤ i ≤ n***

**Output Format**

For each test case print YES if there exists an element in the array, such that the sum of the elements on its left is equal to the sum of the elements on its right; otherwise print NO.

**Sample Input 0**

2

3

1 2 3

4

1 2 3 3

**Sample Output 0**

NO YES

**Explanation 0**

For the first test case, no such index exists.

For the second test case, ***arr[0] + arr[1] = arr[3]***, therefore index ***2*** satisfies the given conditions.

**Sample Input 1**

3

5

1 1 4 1 1

4

2 0 0 0

4

0 0 2 0

**Sample Output 1**

YES YES YES

**Explanation 1**

In the first test case, ***arr[2] = 4*** is between two subarrays summing to ***2***. In the second case, ***arr[0] = 2*** is between two subarrays summing to ***0***. In the third case, ***arr[2] = 2*** is between two subarrays summing to ***0***.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | int main(){ |  |
| 3 | int t,n,Is,rs,m; |
| 4 | scanf("%d",&t); |
| 5 | for(int i=0;i<t;i++){ |
| 6 | Is=0; |
| 7 | rs=0; |
| 8 | scanf("%d",&n); |
| 9 | int arr[n]; |
| 10 | for(int j=0;j<n;j++) |
| 11 | scanf("%d",&arr[j]); |
| 12 | m=n/2; |
| 13 | if(arr[m]==0){ |
| 14 | for(m=0;arr[m]==0 && m<n;m++); |
| 15 | } |
| 16 | for(int j=0;j<=m;j++) |
| 17 | Is=Is+arr[j]; |
| 18 | for(int j=m;j<n;j++) |
| 19 | rs=rs+arr[j]; |
| 20 | printf("%s\n",(Is==rs)?"YES":"NO"); |
| 21 |  |
| 22 | } |
| 23 | return 0; |
| 24 | } |
|  | | | |

**Feedback**

**Input Expected Got**

| 3 |  | |
| --- | --- | --- |
| 5 |
| 1 1 4 | 1 1 YES | YES |
| 4 | YES | YES |
| 2 0 0 | 0 YES | YES |
| 4 |  |  |

2

3

1 2 3

4

1 2 3 3

NO YES

NO YES

REC-First Year-2023

# **GE23131-Programming using C-2023**

**Started on** Sunday, 7 January 2024, 3:29 PM

**State** Finished

**Completed on** Sunday, 7 January 2024, 4:24 PM

**Time taken** 55 mins 8 secs

## **Question 1**

Correct

Marked out of 1.00

**Question text**

Coders here is a simple task for you, you have given an array of size ***N*** and an integer ***M***.

Your task is to calculate the ***difference between maximum sum and minimum sum of N-M*** elements of the given array.

**Constraints:**

***1<=t<=10***

***1<=n<=1000***

***1<=a[i]<=1000***

**Input:**

First line contains an integer ***T*** denoting the number of testcases. First line of every testcase contains two integer ***N*** and ***M***.

Next line contains ***N*** space separated integers denoting the elements of array

**Output:**

For every test case print your answer in new line

SAMPLE INPUT

1

5 1

1 2 3 4 5

SAMPLE OUTPUT

4

Explanation

M is 1 and N is 5 so you have to calculate maximum and minimum sum using (5-1 =) 4 elements. Maximum sum using the 4 elements would be (2+3+4+5=)14.

Minimum sum using the 4 elements would be (1+2+3+4=)10. Difference will be 14-10=4.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | int main() |  |
| 3 | { |
| 4 | int t; |
| 5 | scanf("%d",&t); |
| 6 | while(t--) |
| 7 | { |
| 8 | int n,m,d,min,temp; |
| 9 | scanf("%d %d",&n,&m); |
| 10 | d=n-m; |
| 11 | int arr[n]; |
| 12 | for(int i=0;i<n;i++) |
| 13 | scanf("%d",&arr[i]); |
| 14 | for(int j=0;j<n;j++) |
| 15 | { |
| 16 | min=j; |
| 17 | for(int k=j;k<n;k++) |
| 18 | { |
| 19 | if(arr[k]<arr[min]) |
| 20 | min=k; |
| 21 | } |
| 22 | temp=arr[min]; |
| 23 | arr[min]=arr[j]; |
| 24 | arr[j]=temp; |
| 25 | } |
| 26 | int maxsum=0,minsum=0; |
| 27 | for(int a=0;a<d;a++) |
| 28 | minsum+=arr[a]; |
| 29 | for(int b=n-1;b>m-1;b--) |
| 30 | maxsum+=arr[b]; |
| 31 | printf("%d\n",maxsum-minsum); |
| 32 |  |
| 33 | } |
| 34 |  |
| 35 | } |
|  | | | |

**Feedback**

**Input Expected Got**

1

5 1 4 4

1 2 3 4 5

Passed all tests!

## **Question 2**

Correct

Marked out of 1.00

**Question text**

A new deadly virus has infected large population of a planet. A brilliant scientist has discovered a new strain of virus 77

which can cure this disease. Vaccine produced from this virus has various strength depending on midichlorians count. A person is cured only if midichlorians count in vaccine batch is more than midichlorians count of person. A doctor receives a new set of report which contains midichlorians count of each infected patient, Practo stores all vaccine doctor has and their midichlorians count. You need to determine if doctor can save all patients with the vaccines he has. The number of vaccines and patients are equal.

**Input Format**

First line contains the number of vaccines - N. Second line contains N integers, which are strength of vaccines. Third line contains N integers, which are midichlorians count of patients.

**Output Format**

Print a single line containing ***'Yes'*** or ***'No'***.

**Input Constraint**

***1 < N < 10***

Strength of vaccines and midichlorians count of patients fit in integer.

**SAMPLE INPUT**

5

123 146 454 542 456

100 328 248 689 200

**SAMPLE OUTPUT**

No

Answer:(penalty regime: 0 %)





**Feedback**

**Input Expected Got**

5

123 146 454 542 456 No No

100 328 248 689 200

Passed all tests!

## **Question 3**

Correct

Marked out of 1.00

**Question text**

You are given an array of n integer numbers ***a1, a2, . . . , an***. Calculate the number of pair of indices ***(i, j)*** such that ***1 ≤ i < j ≤ n*** and ***ai*** xor ***aj = 0***.

**Input format**

* First line: ***n*** denoting the number of array elements
* Second line: n space separated integers ***a1, a2, . . . , an***.

**Output format**

Output the required number of pairs.

**Constraints**

***1 ≤ n ≤ 106***

79

***1 ≤ ai ≤ 109***

**SAMPLE INPUT**

5

1 3 1 4 3

**SAMPLE OUTPUT**

2

Explanation

The 2 pair of indices are ***(1, 3)*** and ***(2,5)***.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | int main()  {  int n,count=0; scanf("%d",&n); int arr[n];  for(int i=0;i<n;i++) scanf("%d",&arr[i]); for(int i=0;i<n-1;i++)  {  for(int j=i+1;j<n;j++)  {  if((arr[i]^arr[j])==0) count++;  }  }  printf("%d",count);  } |  |
|  | | | |

**Feedback**

**Input Expected Got**

5 2 2

1 3 1 4 3

Passed all tests!

## **Question 4**

Correct

Marked out of 1.00

**Question text** 80

You are given an array ***A*** of non-negative integers of size ***m***. Your task is to sort the array in non-decreasing order and print out the original indices of the new sorted array.

**Example:**

A={4,5,3,7,1}

After sorting the new array becomes A={1,3,4,5,7}.

The required output should be "4 2 0 1 3"

**INPUT :**

The first line of input consists of the size of the array The next line consists of the array of size m

**OUTPUT :**

Output consists of a single line of integers

**CONSTRAINTS:**

***1<=m<=106***

***0<=A[i]<=106***

NOTE: The indexing of the array starts with 0.

**SAMPLE INPUT**

5

4 5 3 7 1

**SAMPLE OUTPUT**

4 2 0 1 3

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 81 |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8 | int main()  {  int n; scanf("%d",&n); int arr[n];  for(int i=0;i<n;i++) scanf("%d",&arr[i]); |  |



| 9 | } | int max=arr[0]; for(int i=1;i<n;i++)  {  if(arr[i]>max) max=arr[i];  }  max++;  int min=0;  for(int a=0;a<n;a++)  {  for(int b=0;b<n;b++)  {  if(arr[b]<arr[min]) min=b;  }  printf("%d ",min); arr[min]=max;  } |  |  |
| --- | --- | --- | --- | --- |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |
| 19 |
| 20 |
| 21 |
| 22 |
| 23 |
| 24 |
| 25 |
| 26 |
| 27 |
| 28 |
|  | | | | |

**Feedback**

**Input Expected Got**

5 4 2 0 1 3 4 2 0 1 3

4 5 3 7 1

Passed all tests!

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Sunday, 3 January 2025, 10:54 PM

**State** Finished

**Completed on** Sunday, 3 January 2025, 11:23 PM

**Time taken** 29 mins 42 secs

## **Question 1**

Correct

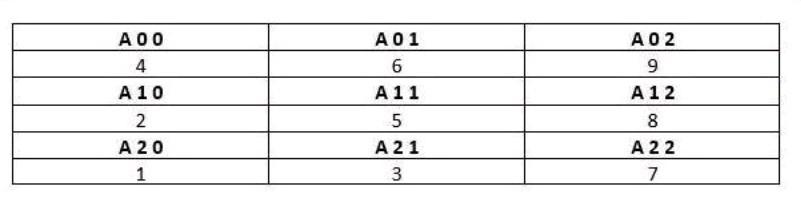
Marked out of 1.00

**Question text**

You are given a two-dimensional 3\*3 array starting from A [0][0]. You should add the alternate elements of the array and print its sum. It should print two different numbers the first being sum of A 0 0, A 0 2, A 1 1, A 2 0, A 2 2 and A 0 1,

A 1 0, A 1 2, A 2 1.

**Input Format**

First and only line contains the value of array separated by single space.



**Output Format**

First line should print sum of A 0 0, A 0 2, A 1 1, A 2 0, A 2 2

Second line should print sum of A 0 1, A 1 0, A 1 2, A 2 1

**SAMPLE INPUT**

1 2 3 4 5 6 7 8 9

**SAMPLE OUTPUT**

25

20

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | int main() |  |
| 3 | { |
| 4 | int arr[3][3]; |
| 5 | for(int i=0;i<3;i++) |
| 6 | { |
| 7 | for(int j=0;j<3;j++) |
| 8 | { |
| 9 | scanf("%d",&arr[i][j]); |
| 10 | } |
| 11 | } |
| 12 | int odd=0,even=0; |
| 13 | for(int i=0;i<3;i++) |
| 14 | { |
| 15 | for(int j=0;j<3;j++) |
| 16 | { |
| 17 | if((i+j)%2!=0) |
| 18 | odd+=arr[i][j]; |
| 19 | else |
| 20 | even+=arr[i][j]; |
| 21 | } |
| 22 | } |
| 23 | printf("%d\n%d",even,odd); |
| 24 |  |
| 25 | } |
|  | | | |

**Feedback**

**Input Expected Got**

1 2 3 4 5 6 7 8 9 25 25

20 20

21 422 423 443 586 645 657 846 904 2591

2356

2591

2356

Passed all tests!

## **Question 2**

Correct

Marked out of 5.00

**Question text**

Microsoft has come to hire interns from your college. N students got shortlisted out of which few were males and a few females. All the students have been assigned talent levels. Smaller the talent level, lesser is your chance to be selected. Microsoft wants to create the result list where it wants the candidates sorted according to their talent levels, but there is a catch. This time Microsoft wants to hire female candidates first and then male candidates.

The task is to create a list where first all-female candidates are sorted in a descending order and then male candidates are sorted in a descending order.

Input Format

The first line contains an integer N denoting the number of students. Next, N lines contain two space-separated integers, ai and bi.

The first integer, ai will be either 1(for a male candidate) or 0(for female candidate).

The second integer, bi will be the candidate's talent level.

Constraints

***1 <= N <= 105***

***0 <= ai <= 1***

***1 <= bi <= 109***

Output Format

Output space-separated integers, which first contains the talent levels of all female candidates sorted in descending order and then the talent levels of male candidates in descending order.

SAMPLE INPUT

5

0 3

1 6

0 2

0 7

1 15

SAMPLE OUTPUT

7 3 2 15 6

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | struct data |  |  |
| 3 | { |  |
| 4 | int gen;int tal; |  |
| 5 |  |  |
| 6 | }; |  |
| 7 | int main() |  |
| 8 | { |  |
| 9 | int n; |  |
| 10 | scanf("%d",&n); |  |
| 11 | struct data a[n]; |  |
| 12 | for(int i=0;i<n;i++) |  |
| 13 | scanf("%d %d",&a[i].gen,&a[i].tal); |  |
| 14 | for(int i=0;i<n-1;i++) |  |
| 15 | { |  |
| 16 | for(int j=0;j<n-i-1;++j) |  |
| 17 | { |  |
| 18 | if(a[j].tal<a[j+1].tal) |  |
| 19 | { |  |
| 20 | struct data temp=a[j]; |  |
| 21 | a[j]=a[j+1]; |  |
| 22 | a[j+1]=temp; |  |
| 23 | } |  |
| 24 | } |  |
| 25 | } |  |
| 26 | for(int i=0;i<n;i++) |  |
| 27 | { |  |
| 28 | if(a[i].gen==0) |  |
| 29  30 | printf("%d ",a[i].tal); | 85 |

| 31  32  33  34  35  36 | }for(int i=0;i<n;++i)  {  if(a[i].gen==1) printf("%d ",a[i].tal);  }  } |  |  |
| --- | --- | --- | --- |
|  | | | |

**Feedback**

| 5 |  |  |
| --- | --- | --- |
| 0 | 3 |
| 1 | 6 | 7 3 2 15 6 |
| 0 | 2 |  |
| 0 | 7 |  |
| 1 | 15 |  |
| 6 |  |  |
| 0 | 1 |  |
| 0 | 26 |  |
| 0 | 39 | 39 37 26 13 7 1 |
| 0 | 37 |  |
| 0 | 7 |  |
| 0 | 13 |  |

**Input Expected Got**

7 3 2 15 6

39 37 26 13 7 1

| 1 | 12 |  |
| --- | --- | --- |
| 1 | 14 |
| 1 | 18 |
| 1 | 1 |
| 1 | 2 |
| 1 | 3 | 31 29 18 14 12 10 9 8 5 3 2 1 |
| 1 | 5 |  |
| 1 | 8 |  |
| 1 | 9 |  |
| 1 | 10 |  |
| 0 | 29 |  |
| 0 | 31 |  |

12

31 29 18 14 12 10 9 8 5 3 2 1

12

| 0 | 12 |  |
| --- | --- | --- |
| 1 | 12 |
| 0 | 12 |
| 1 | 12 |
| 0 | 12 |
| 0 | 12 | 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 |
| 1 | 12 |  |
| 0 | 12 |  |
| 1 | 12 |  |
| 1 | 12 |  |
| 0 | 12 |  |
| 1 | 12 |  |

Passed all tests!

## **Question 3**

Correct

Marked out of 1.00

**Question text**

Shyam Lal, a wealthy landlord from the state of Rajasthan, being an old fellow and tired of doing hard work, decided to sell all his farmland and to live rest of his life with that money. No other farmer is rich enough to buy all his land so he decided to partition the land into rectangular plots of different sizes with different cost per unit area. So, he sold these plots to the farmers but made a mistake. Being illiterate, he made partitions that could be overlapping. When the farmers came to know about it, they ran to him for compensation of extra money they paid to him. So, he decided to return all the money to the farmers of that land which was overlapping with other farmer's land to settle down the conflict. All the portion of conflicted land will be taken back by the landlord.

To decide the total compensation, he has to calculate the total amount of money to return back to farmers with the same cost they had purchased from him. Suppose, Shyam Lal has a total land area of ***1000 x 1000*** equal square blocks where each block is equivalent to a unit square area which can be represented on the co-ordinate axis. Now find the total amount of money, he has to return to the farmers. Help Shyam Lal to accomplish this task.

**Input Format:**

The first line of the input contains an integer ***N***, denoting the total number of land pieces he had distributed. Next ***N*** line contains the ***5*** space separated integers ***(X1, Y1)***, ***(X2, Y2)*** to represent a rectangular piece of land, and cost per unit area ***C***.

***(X1, Y1)*** and ***(X2, Y2)*** are the locations of first and last square block on the diagonal of the rectangular region.

Output Format:

Print the total amount he has to return to farmers to solve the conflict.

Constraints:

***1 ≤ N ≤ 100***

***1 ≤ X1 ≤ X2 ≤ 1000***

***1 ≤ Y1 ≤ Y2 ≤ 1000***

***1 ≤ C ≤ 1000***

SAMPLE INPUT

3

1 4 4 6 1

4 3 6 6 2

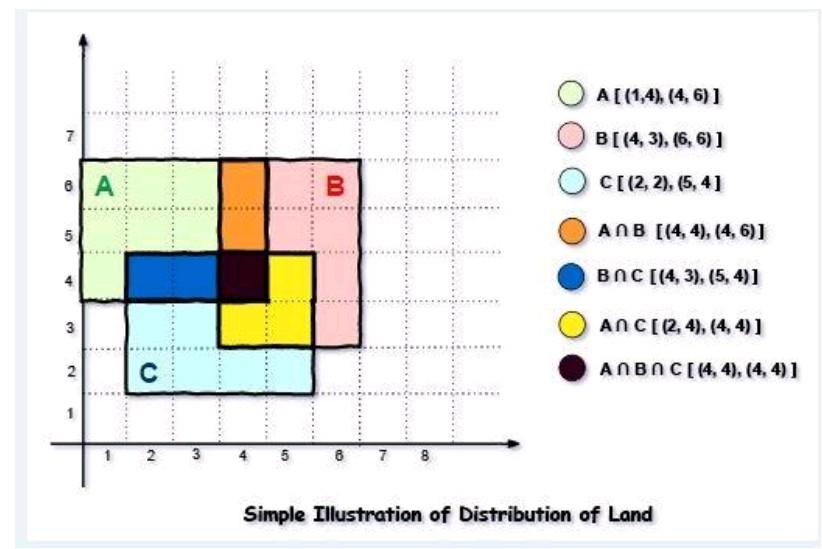
2 2 5 4 3

SAMPLE OUTPUT

35

Explanation



For given sample input (see given graph for reference), compensation money for different farmers is as follows:

Farmer with land area A: C1 = 5 \* 1 = 5 Farmer with land area B: C2 = 6 \* 2 = 12 Farmer with land area C: C3 = 6 \* 3 = 18

Total Compensation Money = C1 + C2 + C3 = 5 + 12 + 18 = 35

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | int main() |  |
| 3 | { |
| 4 | int i,j,n,x1,x2,y1,y2,t=0; |
| 5 | long long total=0; |
| 6 | int arr[1001][1001]={0}; |
| 7 | scanf("%d",&n); |
| 8 | while(n--) |
| 9 | { |
| 10 | scanf("%d %d %d %d %d",&x1,&y1,&x2,&y2,&t); |
| 11 | for(i=x1;i<=x2;i++) |
| 12 | { |
| 13 | for(j=y1;j<=y2;j++) |
| 14 | { |
| 15 | if(arr[i][j]==0) |
| 16 | arr[i][j]+=t; |
| 17 | else if(arr[i][j]>0) |
| 18 | arr[i][j]=(-1)\*(arr[i][j]+t); |
| 19 | else if(arr[i][j]<0) |
| 20 | arr[i][j]-=t; |
| 21 |  |
| 22 | } |
| 23 | } |
| 24 | } |
| 25 | for(i=1;i<1001;i++) |
| 26 | { |
| 27 | for(j=1;j<1001;j++) |
| 28 | { |
| 29 | if(arr[i][j]<0) |
| 30 | total+=arr[i][j]; |
| 31 | } |
| 32 | } |
| 33 | printf("%lld\n",(-1)\*total); |
| 34 | return 0; |
| 35 | } |
|  | | | |

**Input Expected Got**

3

1 4 4 6 1

4 3 6 6 2

2 2 5 4 3

35 35

1

48 12 49 27 8 0 0

3

88 34 99 76 44

82 65 94 100 81

58 16 65 66 7

10500 10500

Passed all tests!

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Sunday, 3 January 2025, 6:46 PM

**State** Finished

**Completed on** Sunday, 3 January 2025, 7:19 PM

**Time taken** 33 mins 53 secs

## **Question 1**

Correct

Marked out of 1.00

**Question text**

Given a string, ***s***, consisting of alphabets and digits, find the frequency of each digit in the given string.

**Input Format**

The first line contains a string, ***num*** which is the given number.

**Constraints**

***1 ≤ len(num) ≤ 1000***

All the elements of num are made of English alphabets and digits.

**Output Format**

Print ten space-separated integers in a single line denoting the frequency of each digit from ***0*** to ***9***.

**Sample Input 0**

a11472o5t6

**Sample Output 0**

0 2 1 0 1 1 1 1 0 0

**Explanation 0**

In the given string:

* ***1*** occurs two times.
* ***2, 4, 5, 6*** and ***7*** occur one time each.



| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | int main()  {  char str[1000]; scanf("%s",str);  int hash[10]={0,0,0,0,0,0,0,0,0,0,};  int temp;  for(int i=0;str[i]!='\0';i++)  {  temp=str[i]-'0'; if(temp<=9&&temp>=0)  {  hash[temp]++;  }  }  for(int i=0;i<=9;i++)  {  printf("%d ",hash[i]);  }  return 0;  } |  |
|  | | | |

The remaining digits ***0, 3, 8*** and ***9*** don't occur at all. Answer:(penalty regime: 0 %)

**Feedback**

**Input Expected Got**

a11472o5t6 0 2 1 0 1 1 1 1 0 0 0 2 1 0 1 1 1 1 0 0

lw4n88j12n1 0 2 1 0 1 0 0 0 2 0 0 2 1 0 1 0 0 0 2 0

1v88886l256338ar0ekk 1 1 1 2 0 1 2 0 5 0 1 1 1 2 0 1 2 0 5 0

Passed all tests!

## **Question 2**

Correct

Marked out of 1.00

**Question text**

Today, Monk went for a walk in a garden. There are many trees in the garden and each tree has an English alphabet on it. While Monk was walking, he noticed that all trees with vowels on it are not in good state. He decided to take care of them. So, he asked you to tell him the count of such trees in the garden.

**Note**: The following letters are vowels: 'A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o' and 'u'.

**Input**:

The first line consists of an integer *T* denoting the number of test cases.

Each test case consists of only one string, each character of string denoting the alphabet (may be lowercase or uppercase) on a tree in the garden.

**Output**:

For each test case, print the count in a new line. 91

**Constraints**:

***1 ≤ T ≤ 10***

***1 ≤ length of string ≤ 105***

**SAMPLE INPUT**

2

nBBZLaosnm JHkIsnZtTL

**SAMPLE OUTPUT**

2

1

**Explanation**

In test case 1, a and o are the only vowels. So, count=2 Answer:(penalty regime: 0 %)

**Feedback**

**Input Expected Got**

2 2 2

nBBZLaosnm 1 1

JHkIsnZtTL

2 2 2

nBBZLaosnm 1 1

JHkIsnZtTL

Passed all tests!

92

## **Question 3**

Correct

Marked out of 1.00

**Question text**

Given a sentence, ***s***, print each word of the sentence in a new line.

**Input Format**

The first and only line contains a sentence, ***s***.

**Constraints**

***1 ≤ len(s) ≤ 1000***

**Output Format**

Print each word of the sentence in a new line.

**Sample Input 0**

This is C

**Sample Output 0**

This is

C

**Explanation 0**



| 1 | #include<stdio.h> |  | 93 |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13  14 | int main()  {  char s[1000]; scanf("%[^\n]s",s);  for(int i=0;s[i]!='\0';i++)  {  if (s[i]!=' ')  printf("%c",s[i]); else  printf("\n");  }  return 0;  } |  |

In the given string, there are three words ["This", "is", "C"]. We have to print each of these words in a new line. Answer:(penalty regime: 0 %)



**Feedback**

| **Input** | **Expected** | **Got** |
| --- | --- | --- |
| This is C | This is C | This is C |
|  | Learning | Learning |

Learning C is fun C

is

fun

C

is fun

Passed all tests!

## **Question 4**

Correct

Marked out of 1.00

**Question text Input Format**

You are given two strings, ***a*** and ***b***, separated by a new line. Each string will consist of lower case Latin characters ('a'- 'z').

**Output Format**

In the first line print two space-separated integers, representing the length of ***a*** and ***b*** respectively. In the second line print the string produced by concatenating ***a*** and ***b*** (***a + b***).

In the third line print two strings separated by a space, ***a'*** and ***b'***. ***a'*** and ***b'*** are the same as ***a*** and ***b***, respectively, except that their first characters are swapped.

**Sample Input**

abcd ef

**Sample Output**

4 2

abcdef ebcd af

a = "abcd"

b = "ef"

|a| = 4

|b| = 2

a + b = "abcdef" a' = "ebcd"

b' = "af"

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | int main() |  |
| 3 | { |
| 4 | char str1[10],str2[10],t; |
| 5 | int i=0,j=0; |
| 6 | int count1=0,count2=0; |
| 7 | scanf("%s",str1); |
| 8 | scanf("%s",str2); |
| 9 | while(str1[i]!='\0') |
| 10 | { |
| 11 | count1++; |
| 12 | i++; |
| 13 |  |
| 14 | } |
| 15 | while(str2[j]!='\0') |
| 16 | { |
| 17 | count2++; |
| 18 | j++; |
| 19 | } |
| 20 | printf("%d %d\n",count1,count2); |
| 21 | printf("%s%s\n",str1,str2); |
| 22 | t=str1[0]; |
| 23 | str1[0]=str2[0]; |
| 24 | str2[0]=t; |
| 25 | printf("%s %s",str1,str2); |
| 26 | return 0; |
| 27 | } |
|  | | | |

**Feedback**

**Input Expected Got**

abcd ef

4 2

abcdef ebcd af

4 2

abcdef ebcd af

Passed all tests!

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Sunday, 3 January 2025, 9:37 PM

**State** Finished

**Completed on** Sunday, 3 January 2025, 10:31 PM

**Time taken** 54 mins 10 secs

## **Question 1**

Correct

Marked out of 1.00

**Question text**

Two strings ***A*** and ***B*** comprising of lower case English letters are compatible if they are equal or can be made equal by following this step any number of times:

· Select a prefix from the string ***A*** (possibly empty), and increase the alphabetical value of all the characters in the prefix by the same valid amount. For example, if the string is ***xyz*** and we select the prefix ***xy*** then we can convert it to ***yx*** by increasing the alphabetical value by 1. But if we select the prefix ***xyz*** then we cannot increase the alphabetical value.

Your task is to determine if given strings ***A*** and ***B*** are compatible.

**Input format**

First line: String ***A***

Next line: String ***B***

**Output format**

For each test case, print ***YES*** if string ***A*** can be converted to string ***B***, otherwise print ***NO***.

Constraints

***1 ≤ len(A) ≤ 1000000***

***1 ≤ len(B) ≤ 1000000***

**SAMPLE INPUT**

abaca cdbda

**SAMPLE OUTPUT**

YES

Explanation

The string ***abaca*** can be converted to ***bcbda*** in one move and to ***cdbda*** in the next move.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | #include<string.h> |  |
| 3 | int main() |
| 4 | { |
| 5 | char str1[1000000],str2[1000000]; |
| 6 | int flag=1; |
| 7 | scanf("%s",str1); |
| 8 | scanf("%s",str2); |
| 9 | int a=strlen(str1); |
| 10 | int b=strlen(str2); |
| 11 | if(a==b) |
| 12 |  |
| 13 | { |
| 14 | for(int i=a-1;i>=0;i--) |
| 15 | { |
| 16 | while(str1[i]!=str2[i]) |
| 17 | { |
| 18 | for(int j=0;j<=i;j++) |
| 19 | { |
| 20 | if(str1[j]<'z') |
| 21 | str1[j]++; |
| 22 | else |
| 23 | { |
| 24 | flag=0; |
| 25 | break; |
| 26 | } |
| 27 | if(flag==0) |
| 28 | break; |
| 29 | } |
| 30 | } |
| 31 | } |
| 32 | } |
| 33 | else |
| 34 | flag=0; |
| 35 |  |
| 36 | if(flag==0) |
| 37 | printf("NO"); |
| 38 | else |
| 39 | printf("YES"); |
| 40 | return 0; |
| 41 | } |
|  | | | |

**Feedback**

**Input Expected Got**

abaca cdbda

YES YES

Passed all tests!

## **Question 2**

Correct

Marked out of 1.00

**Question text**

Danny has a possible list of passwords of Manny's facebook account. All passwords length is odd. But Danny knows that Manny is a big fan of palindromes. So, his password and reverse of his password both should be in the list.

You have to print the length of Manny's password and it's middle character.

**Note: The solution will be unique.**

**INPUT**

The first line of input contains the integer N, the number of possible passwords.

Each of the following N lines contains a single word, its length being an odd number greater than 2 and lesser than ***14***. All characters are lowercase letters of the English alphabet.

**OUTPUT**

The first and only line of output must contain the length of the correct password and its central letter.

**CONSTRAINTS**

***1 ≤ N ≤ 100***

**SAMPLE INPUT**

4

abc def feg cba

**SAMPLE OUTPUT**

3 b

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  | 98 |
| --- | --- | --- | --- |
| 2  3  4  5  6  7  8  9  10  11  12  13 | #include<string.h> int main()  {  int n,flag=0; char temp; scanf("%d",&n);  char words[n][14]; for(int i=0;i<n;i++) scanf("%s",words[i]); char reverse[14]; for(int i=0;i<n-1;i++)  { |  |

| 14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 |  | strcpy(reverse,words[i]); int size=strlen(reverse);  for(int k=0;k<size/2;k++)  {  temp=reverse[k]; reverse[k]=reverse[size-k-1]; reverse[size-k-1]=temp;  }  for(int j=i+1;j<n;j++)  {  if(strcmp(reverse,words[j])==0)  {  flag=1; break;  }  }  if(flag==1) break; |  |  |
| --- | --- | --- | --- | --- |
| }  int len=strlen(reverse);  printf("%d %c ",len,reverse[len/2]); return 0;  } | |
|  | | | | |

**Feedback**

**Input Expected Got**

4

abc def feg cba

3 b 3 b



Passed all tests!

## **Question 3**

Correct

Marked out of 1.00

**Question text**

Joey loves to eat Pizza. But he is worried as the quality of pizza made by most of the restaurants is deteriorating. The last few pizzas ordered by him did not taste good :(. Joey is feeling extremely hungry and wants to eat pizza. But he is confused about the restaurant from where he should order. As always he asks Chandler for help.

Chandler suggests that Joey should give each restaurant some points, and then choose the restaurant having **maximum points**. If more than one restaurant has same points, Joey can choose the one with **lexicographically smallest** name.

Joey has assigned points to all the restaurants, but can't figure out which restaurant satisfies Chandler's criteria. Can you help him out?

**Input:**

First line has N, the total number of restaurants.

Next N lines contain Name of Restaurant and Points awarded by Joey, separated by a space. Restaurant name has **no spaces**, all lowercase letters and will not be more than 20 characters.

**Output:** 99

Print the name of the restaurant that Joey should choose.

**Constraints:**

1 <= N <= 105

1 <= Points <= 106

**SAMPLE INPUT**

3

Pizzeria 108

Dominos 145

Pizzapizza 49

**SAMPLE OUTPUT**

Dominos

**Explanation**

**Dominos** has maximum points.

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | #include<string.h> |  |  |
| 3 | int main() |  |
| 4 | { |  |
| 5 | int n; |  |
| 6 | scanf("%d",&n); |  |
| 7 | char res[n][21]; |  |
| 8 | int rate[n]; |  |
| 9 | for(int i=0;i<n;i++) |  |
| 10 | { |  |
| 11 | scanf("%s",res[i]); |  |
| 12 | scanf("%d",&rate[i]); |  |
| 13 |  |  |
| 14 | } |  |
| 15 | int max=rate[0]; |  |
| 16 | char ans[20]; |  |
| 17 | strcpy(ans,res[0]); |  |
| 18 | for(int i=1;i<n;i++) |  |
| 19 | { |  |
| 20 | if(rate[i]>max) |  |
| 21 | { |  |
| 22 | max=rate[i]; |  |
| 23 | strcpy(ans,res[i]); |  |
| 24 |  |  |
| 25 | } |  |
| 26 | else if(rate[i]==max) |  |
| 27 | { |  |
| 28 | if(strcmp(res[i],ans)<0) |  |
| 29 | strcpy(ans,res[i]); |  |
| 30 | } |  |
| 31 | } |  |
| 32 | printf("%s",ans); | 100 |

| 33  34 | return 0;  } |  |  |
| --- | --- | --- | --- |
|  | | | |

**Feedback**

**Input Expected Got**

3

Pizzeria 108

Dominos 145

Pizzapizza 49

Dominos Dominos

Passed all tests!

## **Question 4**

Correct

Marked out of 1.00

**Question text**

These days Bechan Chacha is depressed because his crush gave him list of mobile number some of them are valid and some of them are invalid. Bechan Chacha has special power that he can pick his crush number only if he has valid set of mobile numbers. Help him to determine the valid numbers.

You are given a string "S" and you have to determine whether it is Valid mobile number or not. Mobile number is valid only if it is of length 10 , consists of numeric values and it shouldn't have prefix zeroes.

**Input:**

First line of input is T representing total number of test cases.

Next T line each representing "S" as described in in problem statement.

**Output:**

Print "YES" if it is valid mobile number else print "NO". Note: Quotes are for clarity.

**Constraints:**

1<= T <= 103

sum of string length <= 105

**SAMPLE INPUT**

3

1234567890

0123456789

0123456.87

101

SAMPLE OUTPUT

YES NO NO

Answer:(penalty regime: 0 %)

| 1 | #include<stdio.h> |  |  |
| --- | --- | --- | --- |
| 2 | #include<string.h> |  |
| 3 | int main() |
| 4 | { |
| 5 | int t; |
| 6 | scanf("%d",&t); |
| 7 | while(t--) |
| 8 | { |
| 9 | int flag=1; |
| 10 | char s[100000]; |
| 11 | scanf("%s",s); |
| 12 | int k=strlen(s); |
| 13 |  |
| 14 | if(k==10) |
| 15 | { |
| 16 | for(int i=0;i<10;i++) |
| 17 | { |
| 18 | if(s[0]=='0') |
| 19 | { |
| 20 | flag=0; |
| 21 | break; |
| 22 | } |
| 23 | if(s[i]<'0'||s[i]>'9') |
| 24 | { |
| 25 | flag=0; |
| 26 | break; |
| 27 | } |
| 28 | } |
| 29 | } |
| 30 | else |
| 31 | flag=0; |
| 32 | if(flag==1) |
| 33 | printf("YES\n"); |
| 34 | else |
| 35 | printf("NO\n"); |
| 36 | } |
| 37 | return 0; |
| 38 |  |
| 39 | } |
|  | | | |

**Feedback**

**Input Expected Got**

3 YES

1234567890 NO

0123456789 NO

0123456.87

YES NO NO

Passed all tests!

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Sunday, 3 January 2025, 11:25 PM

**State** Finished

**Completed on** Sunday, 3 January 2025, 11:29 PM

**Time taken** 4 mins 18 secs

## **Question 1**

Correct

Marked out of 1.00

**Question text**

A binary number is a combination of 1s and 0s. Its nth least significant digit is the nth digit starting from the right starting with 1. Given a decimal number, convert it to binary and determine the value of the the 4th least significant digit.

**Example**

number = 23

* Convert the decimal number 23 to binary number: 2310 = 24 + 22 + 21 + 20 = (10111)2.
* The value of the 4th index from the right in the binary representation is 0.

**Function Description**

Complete the function fourthBit in the editor below.

fourthBit has the following parameter(s):

int number: a decimal integer

Returns:

int: an integer 0 or 1 matching the 4th least significant digit in the binary representation of number.

**Constraints**

0 ≤ number < 231

**Input Format for Custom Testing**

Input from stdin will be processed as follows and passed to the function.

103

The only line contains an integer, number.

**Sample Case 0**

**Sample Input 0**

STDIN Function



32 → number = 32

**Sample Output 0**

0

**Explanation 0**

* Convert the decimal number 32 to binary number: 3210 = (100000)2.
* The value of the 4th index from the right in the binary representation is 0.

**Sample Case 1**

**Sample Input 1**

STDIN Function



77 → number = 77

**Sample Output 1**

1

**Explanation 1**

* Convert the decimal number 77 to binary number: 7710 = (1001101)2.
* The value of the 4th index from the right in the binary representation is 1.

Answer:(penalty regime: 0 %)

| 1 | /\* |  | 104 |
| --- | --- | --- | --- |
| 2  3  4  5 | * Complete the 'fourthBit' function below.   \*   * The function is expected to return an INTEGER. * The function accepts INTEGER number as parameter. |  |



| 6 | \*/ |  |  |
| --- | --- | --- | --- |
| 7 |  |
| 8 | int fourthBit(int number) |
| 9 | { |
| 10 | int binary[32]; |
| 11 | int i=0; |
| 12 | while(number>0) |
| 13 | { |
| 14 | binary[i]=number%2; |
| 15 | number/=2; |
| 16 | i++; |
| 17 | } |
| 18 | if(i>=4) |
| 19 | { |
| 20 | return binary[3]; |
| 21 |  |
| 22 | } |
| 23 | else |
| 24 | return 0; |
| 25 | } |
|  | | | |

**Feedback**

**Test Expected Got**

printf("%d", fourthBit(32)) 0 0

printf("%d", fourthBit(77)) 1 1

Passed all tests!

## **Question 2**

Correct

Marked out of 1.00

**Question text**

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the pth element of the list, sorted ascending. If there is no pth element, return 0.

**Example**

n = 20

p = 3

The factors of 20 in ascending order are {1, 2, 4, 5, 10, 20}. Using 1-based indexing, if p = 3, then 4 is returned. If p >

6, 0 would be returned.

**Function Description**

Complete the function pthFactor in the editor below.

pthFactor has the following parameter(s):

int n: the integer whose factors are to be found int p: the index of the factor to be returned

Returns:

int: the long integer value of the pth integer factor of n or, if there is no factor at that index, then 0 is returned

**Constraints**

1 ≤ n ≤ 1015

1 ≤ p ≤ 109

Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer n, the number to factor.

The second line contains an integer p, the 1-based index of the factor to return.

**Sample Case 0**

**Sample Input 0**

STDIN Function



10 → n = 10

3 → p = 3

**Sample Output 0**

5

**Explanation 0**

Factoring n = 10 results in {1, 2, 5, 10}. Return the p = 3rd factor, 5, as the answer.

**Sample Case 1**

**Sample Input 1**

STDIN Function



10 → n = 10

5 → p = 5

**Sample Output 1**

0

**Explanation 1**

Factoring n = 10 results in {1, 2, 5, 10}. There are only 4 factors and p = 5, therefore 0 is returned as the answer.

**Sample Case 2**

**Sample Input 2**

STDIN Function



| 1 | → | n = 1 |
| --- | --- | --- |
| 1 | → | p = 1 |

**Sample Output 2**

1

**Explanation 2**

Factoring n = 1 results in {1}. The p = 1st factor of 1 is returned as the answer.

Answer:(penalty regime: 0 %)

| 1 | /\* |  |  |
| --- | --- | --- | --- |
| 2 | \* Complete the 'pthFactor' function below. |  |
| 3 | \* |
| 4 | \* The function is expected to return a LONG\_INTEGER. |
| 5 | \* The function accepts following parameters: |
| 6 | \* 1. LONG\_INTEGER n |
| 7 | \* 2. LONG\_INTEGER p |
| 8 | \*/ |
| 9 |  |
| 10 | long pthFactor(long n, long p) |
| 11 | { |
| 12 | int count=0; |
| 13 | for(long i=1;i<=n;++i) |
| 14 | { |
| 15 | if(n%i==0) |
| 16 | { |
| 17 | count++; |
| 18 | if(count==p) |
| 19 | { |
| 20 | return i; |
| 21 | } |
| 22 | } |
| 23 | } |
| 24 | return 0; |
| 25 | } |
|  | | | |

**Test Expected Got**

| printf("%ld", pthFactor(10, 3)) 5 | 5 |
| --- | --- |
| printf("%ld", pthFactor(10, 5)) 0 | 0 |
| printf("%ld", pthFactor(1, 1)) 1 | 1 |

Passed all tests!

## **Question 3**

Correct

Marked out of 1.00

**Question text**

Find the number of ways that a given integer, X, can be expressed as the sum of the Nth powers of unique, natural numbers.

For example, if X = 13 and N = 2, we have to find all combinations of unique squares adding up to 13. The only solution is 22 + 32.

**Function Description**

Complete the powerSum function in the editor below. It should return an integer that represents the number of possible combinations.

powerSum has the following parameter(s): X: the integer to sum to

N: the integer power to raise numbers to

Input Format

The first line contains an integer X. The second line contains an integer N.

**Constraints**

1 ≤ X ≤ 1000

2 ≤ N ≤ 10

**Output Format**

Output a single integer, the number of possible combinations calculated.

**Sample Input**

10

2

**Sample Output**

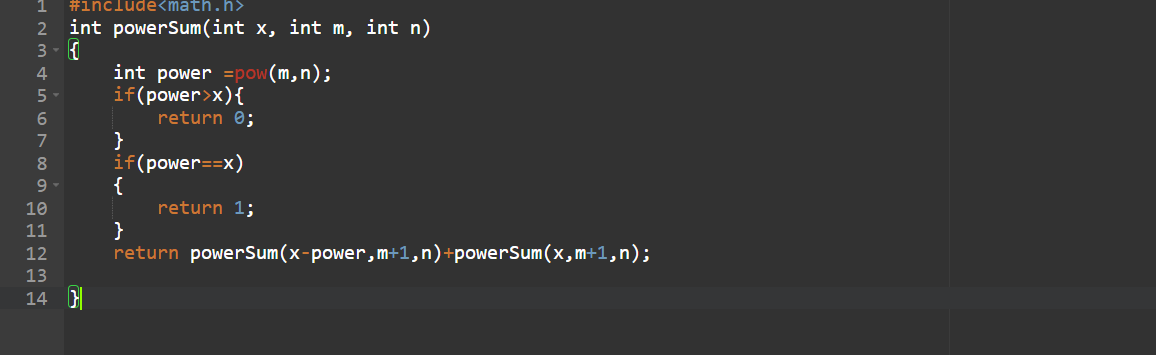
1

**Explanation**

If X = 10 and N = 2, we need to find the number of ways that 10 can be represented as the sum of squares of unique numbers.

10 = 12 + 32

This is the only way in which 10 can be expressed as the sum of unique squares.



## **Question 4**

Correct

Marked out of 1.00

**Question text**

You are a bank account hacker. Initially you have 1 rupee in your account, and you want exactly N rupees in your account. You wrote two hacks, first hack can multiply the amount of money you own by 10, while the second can multiply it by 20. These hacks can be used any number of time. Can you achieve the desired amount N using these hacks.

**Constraints:**

1<=T<=100

1<=N<=10^12

**Input**

* The test case contains a single integer N.

**Output**

For each test case, print a single line containing the string "1" if you can make exactly N rupees or "0" otherwise.

**SAMPLE INPUT**

1

**SAMPLE OUTPUT**

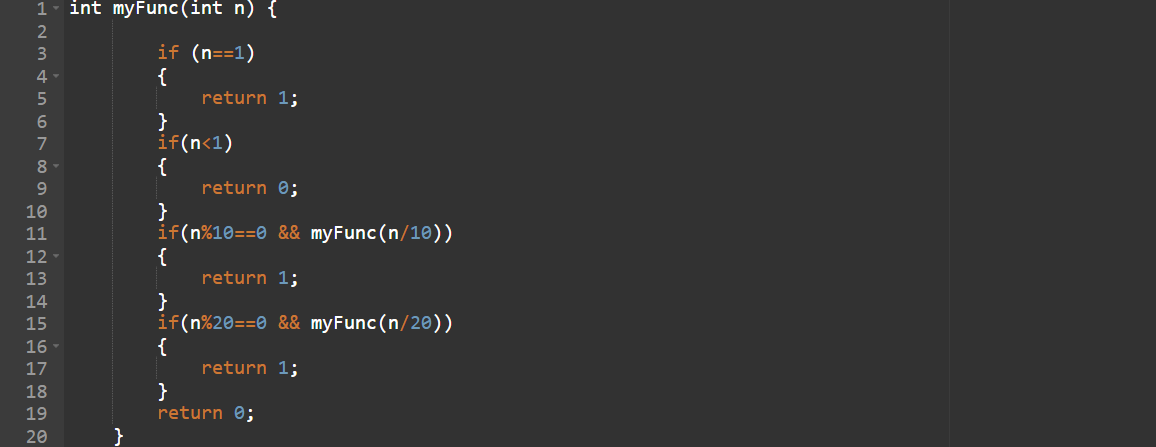
1

**SAMPLE INPUT**

2

**SAMPLE OUTPUT**

0

****

REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Sunday, 4 January 2025, 11:25 PM

**State** Finished

**Completed on** Sunday, 4 January 2025, 11:29 PM

**Time taken** 4 mins 18 secs

## **Question 1**

Correct

Marked out of 1.00

**Question text**

Given an array of numbers, find the index of the smallest array element (the pivot), for which the sums of all elements to the left and to the right are equal. The array may not be reordered.

**Example**:

arr=[1,2,3,4,6]

* the sum of the first three elements, 1+2+3=6. The value of the last element is 6.
* Using zero based indexing, arr[3]=4 is the pivot between the two subarrays.
* The index of the pivot is 3.

**Function Description:**

Complete the function balancedSum in the editor below. balancedSum has the following parameter(s): int arr[n]: an array of integers Returns: int: an integer representing the index of the pivot

**Constraints**

* 3 ≤ n ≤ 105
* 1 ≤ arr[i] ≤ 2 × 104, where 0 ≤ I < n
* It is guaranteed that a solution always exists.

**Input Format for Custom Testing**

Input from stdin will be processed as follows and passed to the function. The first line contains an integer n, the size of the array arr. Each of the next n lines contains an integer, arr[i], where 0 ≤ i < n.

**Sample Input**

STDIN Function Parameters

4 → arr[] size n = 4

1 → arr = [1, 2, 3, 3]

2

3

3

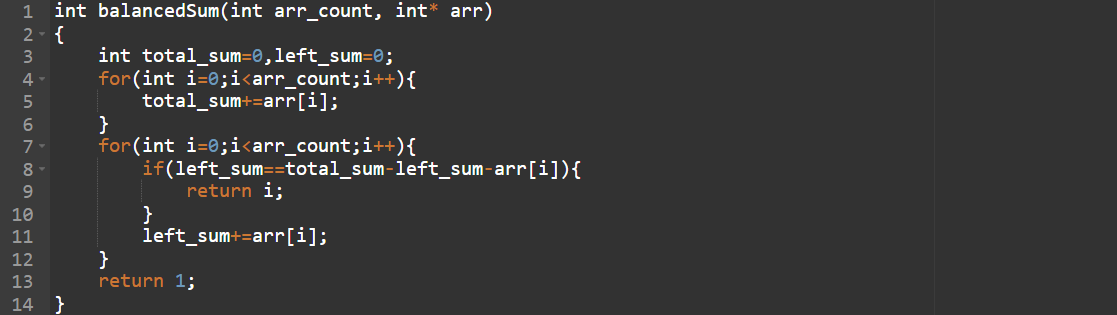
**Sample Output** 0

2

**Explanation 0**

* The sum of the first two elements, 1+2=3. The value of the last element is 3.
* Using zero based indexing, arr[2]=3 is the pivot between the two subarrays.

The index of the pivot is 2.



## **Question 2**

Correct

Marked out of 1.00

**Question text**

Calculate the sum of an array of integers.

**Example**

numbers = [3, 13, 4, 11, 9]

The sum is 3 + 13 + 4 + 11 + 9 = 40.

**Function Description**

Complete the function arraySum in the editor below.

arraySum has the following parameter(s): int numbers[n]: an array of integers

Returns

int: integer sum of the numbers array

**Constraints**

1 ≤ n ≤ 104

1 ≤ numbers[i] ≤ 104

**Input Format for Custom Testing**

Input from stdin will be processed as follows and passed to the function. The first line contains an integer n, the size of the array numbers.

Each of the next n lines contains an integer numbers[i] where 0 ≤ i < n.

**Sample Input**

STDIN Function

5 → numbers[] size n = 5

1 → numbers = [1, 2, 3, 4, 5]

2

3

4

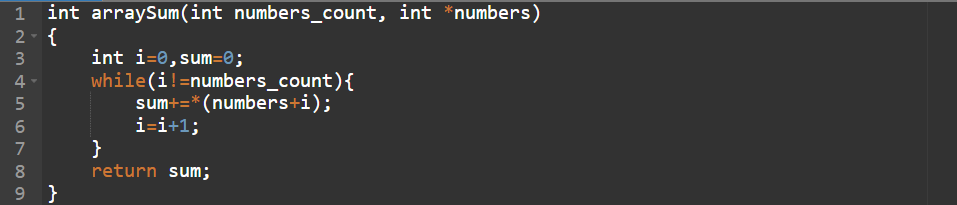
5

**Sample Output**

15

**Explanation**

1 + 2 + 3 + 4 + 5 = 15.



## **Question 3**

Correct

Marked out of 1.00

**Question text**

Given an array of n integers, rearrange them so that the sum of the absolute differences of all adjacent elements is minimized. Then, compute the sum of those absolute differences.

**Example**

n = 5, arr = [1, 3, 3, 2, 4]

If the list is rearranged as arr' = [1, 2, 3, 3, 4], the absolute differences are |1 - 2| = 1, |2 - 3| = 1,|3 - 3| = 0, |3 - 4| = 1. The sum of those differences is 1 + 1 + 0 + 1 = 3.

**Function Description**

Complete the function minDiff in the editor below.

minDiff has the following parameter: arr: an integer array

Returns:

int: the sum of the absolute differences of adjacent elements

Constraints

2 ≤ n ≤105

0 ≤ arr[i] ≤ 109, where 0 ≤ i < n

**Input Format For Custom Testing**

The first line of input contains an integer, n, the size of arr.

Each of the following n lines contains an integer that describes arr[i] (where 0 ≤ i < n) .

**Sample Input For Custom Testing STDIN Function**

5 → arr[] size n = 5

5 → arr[] = [5, 1, 3, 7, 3]

1

3

7

3

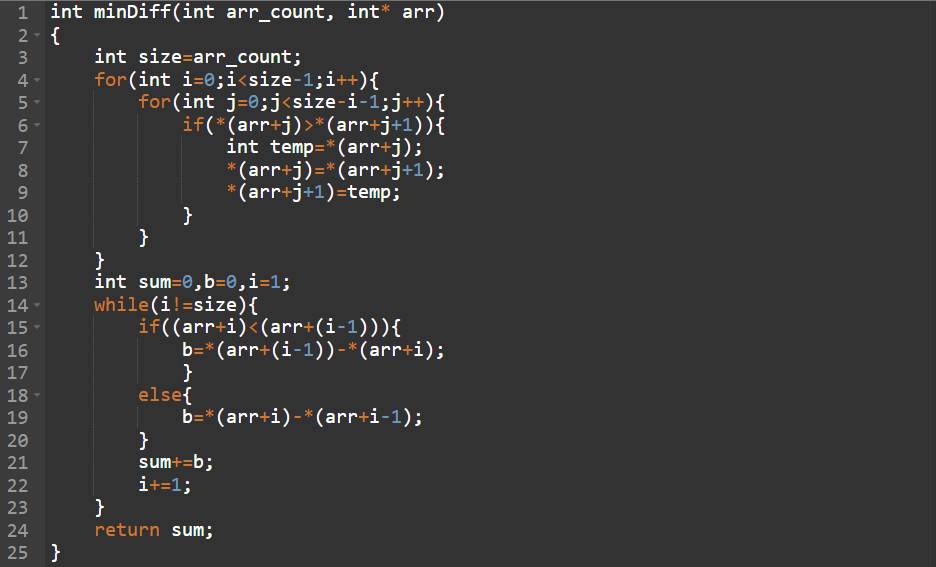
**Sample Output**

6

**Explanation**

n = 5, arr = [5, 1, 3, 7, 3]

If arr is rearranged as arr' = [1, 3, 3, 5, 7], the differences are minimized. The final answer is |1 - 3| + |3 - 3| + |3 - 5| + |5 - 7| = 6.



REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Sunday, 7 January 2025, 10:54 PM

**State** Finished

**Completed on** Sunday, 7 January 2025, 11:23 PM

**Time taken** 29 mins 42 secs

## **Question 1**

Correct

Marked out of 1.00

**Question text**

You are transporting some boxes through a tunnel, where each box is a parallelepiped, and is characterized by its length, width and height.

The height of the tunnel 41 feet and the width can be assumed to be infinite. A box can be carried through the tunnel only if its height is strictly less than the tunnel's height. Find the volume of each box that can be successfully transported to the other end of the tunnel.

Note: Boxes cannot be rotated.

**Input Format**

The first line contains a single integer n, denoting the number of boxes.

n lines follow with three integers on each separated by single spaces - lengthi, widthi and heighti which are length, width and height in feet of the i-th box.

Constraints

1 ≤ n ≤ 100

1 ≤ lengthi, widthi, heighti ≤ 100

Output Format

For every box from the input which has a height lesser than 41 feet, print its volume in a separate line.

**Sample Input**

4

5 5 5

1 2 40

10 5 41

7 2 42

**Sample Output**

125

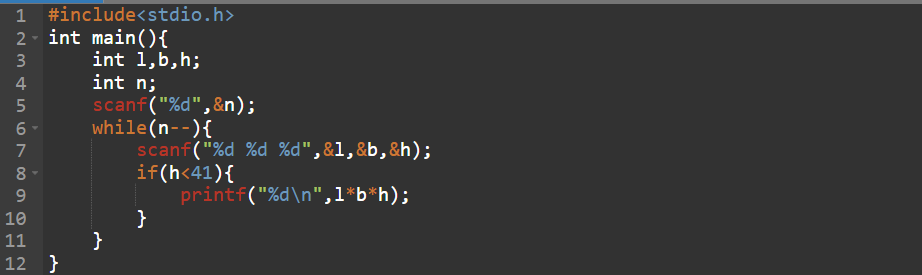
80

**Explanation**

The first box is really low, only 5 feet tall, so it can pass through the tunnel and its volume is 5 x 5 x 5 = 125.

The second box is sufficiently low, its volume is 1 x 2 x 4= = 80.

The third box is exactly 41 feet tall, so it cannot pass. The same can be said about the fourth box.



## **Question 2**

Correct

Marked out of 1.00

**Question text**

You are given n triangles, specifically, their sides ai, bi and ci. Print them in the same style but sorted by their areas from the smallest one to the largest one. It is guaranteed that all the areas are different.

The best way to calculate a volume of the triangle with sides a, b and c is Heron's formula: S = Ö p \* (p – a) \* (p - b) \* (p – c) where p = (a + b + c) / 2.

**Input Format**

First line of each test file contains single integer n. n lines follow with ai, bi and ci on each separated by single spaces.

**Constraints**

1 ≤ n ≤ 100

1 ≤ ai, bi, ci ≤ 70

ai + bi > ci, ai + ci > bi and bi + ci > ai

**Output Format**

Print exactly n lines. On each line print 3 integers separated by single spaces, which are ai, bi and ci of the corresponding triangle.

**Sample Input**

3

7 24 25

5 12 13

3 4 5

**Sample Output**

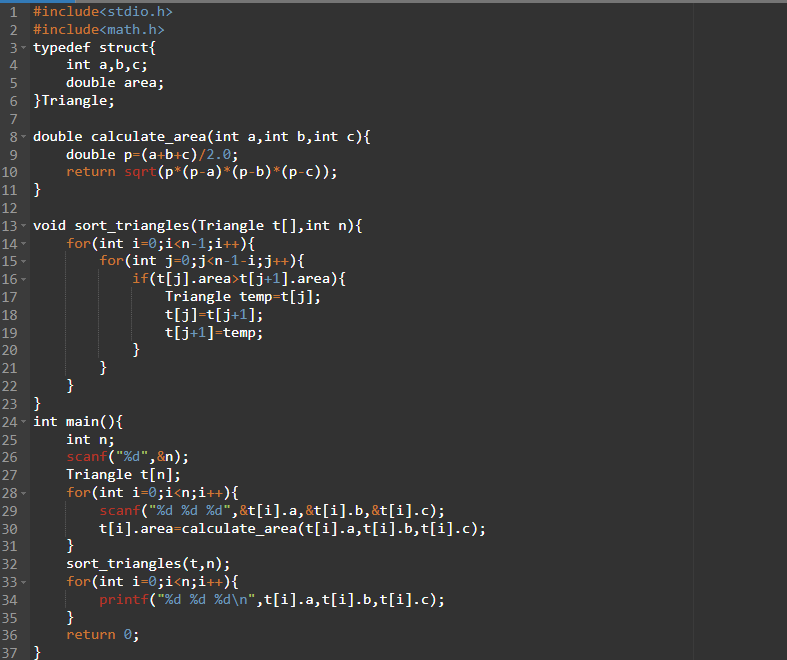
3 4 5

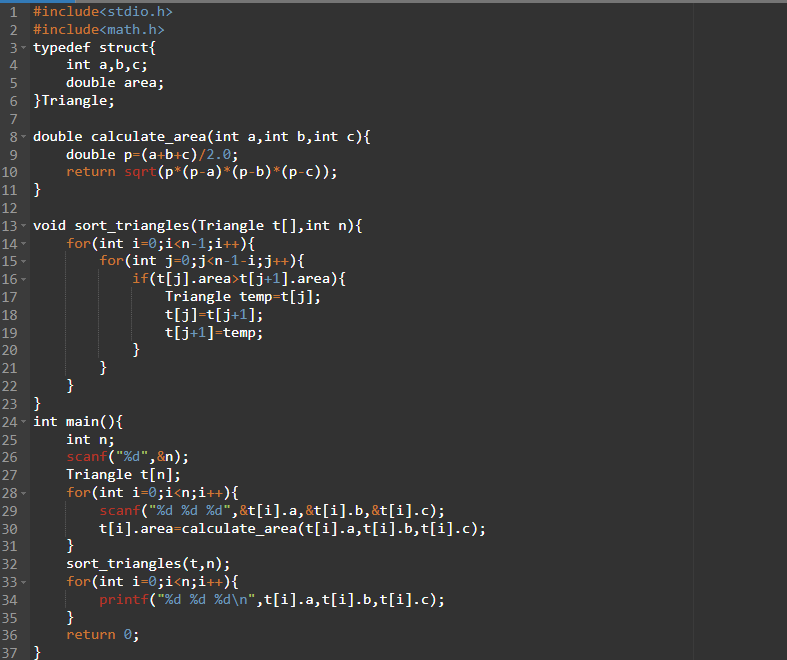
5 12 13

7 24 25

**Explanation**

The square of the first triangle is 84. The square of the second triangle is 30. The square of the third triangle is 6. So, the sorted order is the reverse one.





REC-First Year-2024

# **GE23131-Programming using C-2024**

**Started on** Sunday, 7 January 2025, 11:30 PM

**State** Finished

**Completed on** Sunday, 7 January 2025, 12:02 PM

**Time taken** 31 mins 44 secs

## **Question 1**

Correct

Marked out of 1.00

**Question text**

Given an array of integers, reverse the given array in place using an index and loop rather than a built-in function.

**Example**

arr = [1, 3, 2, 4, 5]

Return the array [5, 4, 2, 3, 1] which is the reverse of the input array.

**Function Description**

Complete the function reverseArray in the editor below.

reverseArray has the following parameter(s): int arr[n]: an array of integers

Return

int[n]: the array in reverse order

**Constraints**

1 ≤ n ≤ 100

0 < arr[i] ≤ 100

**Input Format For Custom Testing**

The first line contains an integer, n, the number of elements in arr.

Each line i of the n subsequent lines (where 0 ≤ i < n) contains an integer, arr[i].

**Sample Input For Custom Testing**

5

1

3

2

4

5

**Sample Output**

5

4

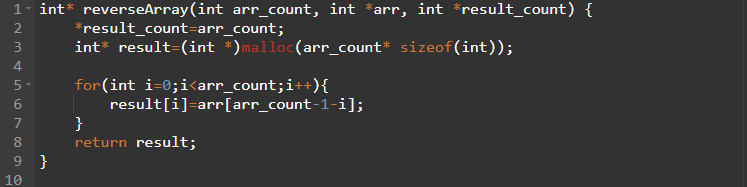
2

3

1

**Explanation**

The input array is [1, 3, 2, 4, 5], so the reverse of the input array is [5, 4, 2, 3, 1].



## **Question 2**

Correct

Marked out of 1.00

**Question text**

An automated cutting machine is used to cut rods into segments. The cutting machine can only hold a rod of minLength or more, and it can only make one cut at a time. Given the array lengths [] representing the desired lengths of each segment, determine if it is possible to make the necessary cuts using this machine. The rod is marked into lengths already, in the order given.

**Example**

n = 3

lengths = [4, 3, 2]

minLength = 7

The rod is initially sum(lengths) = 4 + 3 + 2 = 9 units long. First cut off the segment of length 4 + 3 = 7 leaving a rod 9 - 7 = 2. Then check that the length 7 rod can be cut into segments of lengths 4 and 3. Since 7 is greater than or equal to minLength = 7, the final cut can be made. Return "Possible".

**Example**

n = 3

lengths = [4, 2, 3]

minLength = 7

The rod is initially sum(lengths) = 4 + 2 + 3 = 9 units long. In this case, the initial cut can be of length 4 or 4 + 2 = 6. Regardless of the length of the first cut, the remaining piece will be shorter than minLength. Because n - 1 = 2 cuts cannot be made, the answer is "Impossible".

**Function Description**

Complete the function cutThemAll in the editor below.

cutThemAll has the following parameter(s):

int lengths[n]: the lengths of the segments, in order

int minLength: the minimum length the machine can accept

Returns

string: "Possible" if all n-1 cuts can be made. Otherwise, return the string "Impossible".

**Constraints**

**SSS**

* 2 ≤ n ≤ 105
* 1 ≤ t ≤ 109
* 1 ≤ lengths[i] ≤ 109

The sum of the elements of lengths equals the uncut rod length.

Input Format For Custom Testing

The first line contains an integer, n, the number of elements in lengths.

Each line i of the n subsequent lines (where 0 ≤ i < n) contains an integer, lengths[i]. The next line contains an integer, minLength, the minimum length accepted by the machine.

**Sample Input For Custom Testing**

| STDIN |  | Function |
| --- | --- | --- |
| 4 | → | lengths[] size n = 4 |
| 3 | → | lengths[] = [3, 5, 4, 3] |
| 5 |  |  |
| 4 |  |  |
| 3 |  |  |
| 9 | → | minLength= 9 |

**Sample Output**

Possible

**Explanation**

The uncut rod is 3 + 5 + 4 + 3 = 15 units long. Cut the rod into lengths of 3 + 5 + 4

= 12 and 3.

Then cut the 12-unit piece into lengths 3 and 5 + 4 = 9.

The remaining segment is 5 + 4 = 9 units and that is long enough to make the final cut.

