Week 1-02

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 %)

```c
#include<stdio.h>
int main ()
{
  char a;
  scanf("%c",&a);
  printf("%c\n",a);
  int x,y,z;
  scanf("%d %d %d",&x,&y,&z);
  printf("%d",(x+y+z)/3);
  return 0;
}
```

	Input	Expected	Got	
☐	A 3 4 6	A 4	A 4	☐
☐	T 7 3 8	T 6	T 6	☐
☐	R 0 100 99	R 66	R 66	☐

Passed all tests! ☐

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 %)

```c
#include<stdio.h>
int main()
{int a;
long l;
char ch;
float f;
double lf;
scanf("%d %ld %c %f %lf",&a,&l,&ch,&f,&lf);
printf("%d\n%ld\n%c\n%.3f\n%.9lf",a,l,ch,f,lf);
return 0;
}
```

	Input	Expected	Got	
✓	3 12345678912345 a 334.23 14049.30493	3 12345678912345 a 334.230 14049.304930000	3 12345678912345 a 334.230 14049.304930000	✓

Passed all tests! ✓

Write a program to print the ASCII value and the two adjacent characters of the given character.

Input

E

Output

69

D F

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{char ch;
scanf("%c",&ch);
printf("%d",ch);
printf("\n%c %c",ch-1, ch+1);
return 0;}
```

	Input	Expected	Got	
✓	E	69 D F	69 D F	✓

Passed all tests! ✓

Week 1-01

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Saturday, 21 December 2024, 11:42 AM
Duration	2 days 5 hours

Question **1**

Correct

Marked out of 3.00

Flag question

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

Hello, World!

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    printf("Hello, World!");
}
```

	Expected	Got	
✓	Hello, World!	Hello, World!	✓

Passed all tests! ✓

Question 2
Correct
Marked out of 5.00
Flag question

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 %)

```
#include <stdio.h>
int main()
{
    char ch;
    scanf("%c",&ch);
    printf("%c",ch);
}
```

	Input	Expected	Got	
✓	c	c	c	✓

Passed all tests! ✓

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 %)

```c
#include<stdio.h>
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);
}
```

	Input	Expected	Got	
✓	10 4 4.0 2.0	14 6 6.0 2.0	14 6 6.0 2.0	✓
✓	20 8 8.0 4.0	28 12 12.0 4.0	28 12 12.0 4.0	✓

Passed all tests! ✓

Week 2-01

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 %)

```c
#include<stdio.h>
int main()
{int a,b;
scanf("%d%d",&a,&b);
float c,d,e;
c=(a*12)*2.54;
d=b*2.54;
e=c+d;
printf("%.2f",e);
return 0;
}
```

	Input	Expected	Got	
✓	5 6	167.64	167.64	✓

Passed all tests! ✓

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 %)

```c
#include<stdio.h>
int main()
{int a,b;
scanf("%d%d",&a,&b);
printf("%d",a+b);
printf("\n%d",a-b);
printf("\n%d",a*b);
printf("\n%d",a/b);
printf("\n%d",a%b);
return 0;
}
```

	Input	Expected	Got	
✓	100 6	106 94 600 16 4	106 94 600 16 4	✓

Passed all tests! ✓

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 %)

```c
#include<stdio.h>
int main()
{
    int a;
scanf("%d",&a);
float Regularprice, Discount, Total;
scanf("%f%f%f",&Regularprice,&Discount,&Total);
Regularprice=a*3.49;
Discount=Regularprice*0.60;
Total=Regularprice-Discount;
printf("Regular price: %.2f", Regularprice);
printf("\nDiscount: %.2f",Discount);
printf("\nTotal: %.2f",Total);
return 0;
}
```

	Input	Expected	Got	
✓	10	Regular price: 34.90 Discount: 20.94 Total: 13.96	Regular price: 34.90 Discount: 20.94 Total: 13.96	✓

Passed all tests! ✓

Week 2-02

leted Friday, 25 October 2024, 11:29 AM

ation 59 days 6 hours

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 %)

```c
#include<stdio.h>
int main()
{
    int x,y;
    scanf("%d%d",&x,&y);
    if (1<=x && y<=1000000)
    {
        if(y>=x)
        {printf("YES");
        }
        else
        {
            printf("NO");
        }
    }
        return 0;
}
```

	Input	Expected	Got	
✓	100 110	YES	YES	✓
✓	100 90	NO	NO	✓

Passed all tests! ✓

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 exact
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 %)

```c
#include<stdio.h>
int main()
{
    int n,hand;
    scanf("%d",&n);
    hand=n*(n-1)/2;
    printf("%d", hand);
    return 0;
}
```

	Input	Expected	Got	
✓	1	0	0	✓
✓	2	1	1	✓

Passed all tests! ✓

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 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

Explanation Out of given numbers, 8 is maximum.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int a,b,c;
    scanf("%d%d%d",&a,&b,&c);
    if(a>b && a>c)
    {
        printf("%d",a);
    }
    else if(b>c && b>a)
    {
        printf("%d",b);
    }
    else
    {
        printf("%d",c);
    }
    return 0;
}
```

	Input	Expected	Got	
✓	81 26 15	81	81	✓

Passed all tests! ✓

Week 3-02

Sample Output 3

The number of sides is not supported.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main ()
{
    int a;
    scanf("%d",&a);
    switch(a)
    {
        case 3:
        printf("Triangle");
        break;
        case 4:
        printf("Square");
        break;
        case 5:
        printf("Pentagon");
        break;
        case 6:
        printf("Hexagon");
        break;
        case 7:
        printf("Heptagon");
        break;
        case 8:
        printf ("Octagon");
        break;
        case 9:
        printf("Nonagon");
        break;
        case 10:
        printf("Decagon");
        break;
        default:
        printf("The number of sides is not supported.");
    }
        return 0;
}
```

	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! ✓

Answer: (penalty regime: 0 %)

```c
{
    int n;
    scanf("%d",&n);
    if(n%12==8)
    {
        printf("Dragon");
    }
    else if(n%12==9)
    {
        printf("Snake");
    }
    else if(n%12==10)
    {
        printf("Horse");
    }
    else if (n%12==11)
    {
        printf("Sheep");
    }
    else if (n%12==0)
    {
        printf("Monkey");
    }
    else if (n%12==1)
    {
        printf("Rooter");
    }
    else if (n%12==2)
    {
        printf("Dog");
    }
    else if(n%12==3)
    {
        printf("Pig");
    }
    else if (n%12==4)
    {
        printf("Rat");
    }
    else if(n%12==5)
    {
        printf("Ox");
    }
    else if (n%12==6)
    {
        printf("Tiger");
    }
    else
    {
        printf("Hare");
    }
}
```

	Input	Expected	Got	
✓	2004	Monkey	Monkey	✓
✓	2010	Tiger	Tiger	✓

Passed all tests! ✓

will always be entered. It does not need to perform any error checking.

Input 1

Output 1

are is black.

Input 2

Output 2

are is white.

: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    char column;
    int row;
    scanf("%c %d",&column,&row);
    int is_black_start=(column-'a')%2==0;
    if((is_black_start&&row%2!=0)||(!is_black_start&&row%2==0))
    {
        printf("The square is black.\n");
    }
    else
    {
        printf("The square is white.\n");
    }
}
```

Input	Expected	Got	
a 1	The square is black.	The square is black.	✓
d 5	The square is white.	The square is white.	✓

all tests! ✓

Week 3-01

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 b

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n,m;
    scanf("%d %d",&n,&m);
    if(n%10==m%10)
    {
        printf("true");
    }
    else
    {
        printf("false");
    }
}
```

	Input	Expected	Got	
✓	25 53	false	false	✓
✓	27 77	true	true	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    if(n%2 == 0)
    {
       if(n>=2 || n<=5)
       {
           printf("Not Weird");
       }

       else if(n>=6 || n<=20)
       {
           printf("Weird");
       }

       else if(n>20)
       {
           printf("Not Weird");
       }
    }
    else
    {
        printf("Weird");
    }

}
```

	Input	Expected	Got	
✓	3	Weird	Weird	✓
✓	24	Not Weird	Not Weird	✓

Passed all tests! ✓

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
Input 1 3 5 4 Sample Output 1 yes Sample Input 2 5 8 2 Sample Output 2 no

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int a,b,c;
    scanf("%d %d %d",&a,&b,&c);
    if(a*a == b*b + c*c || b*b == a*a + c*c || c*c == a*a + b*b )
    {
        printf("yes");
    }
    else
    {
        printf("no");
    }

}
```

	Input	Expected	Got	
✓	3 5 4	yes	yes	✓
✓	5 8 2	no	no	✓

Passed all tests! ✓

Week 3-03

6

2020

Sample Output 1

170

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int day,mon,yr,is_leap;
    scanf("%d %d %d",&day,&mon,&yr);
    if(((yr%4==0)&&(yr%100!=0))||(yr%400==0))
    is_leap=1;
    {
        if(mon>1)
        day+=31;
        if(mon>2)
        day+=is_leap?29:28;
        if(mon>3)
        day+=31;
        if(mon>4)
        day+=30;
        if(mon>5)
        day+=31;
        if(mon>6)
        day+=30;
        if(mon>7)
        day+=31;
        if(mon>8)
        day+=31;
        if(mon>9)
        day+=30;
        if(mon>10)
        day+=31;
        if(mon>11)
        day+=30;
    }
    printf("%d",day);
}
```

	Input	Expected	Got	
✓	18 6 2020	170	170	✓

Passed all tests! ✓

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 %)

```c
#include<stdio.h>
int main()
{
    char shape;
    int side1,side2;
    scanf("%c",&shape);
    scanf("%d %d",&side1,&side2);
    if(shape=='R')
    {
        printf("%d\n",(side1*side2));
    }
    else if(shape=='S')
    {
        printf("%d\n",(side1*side2)/2);
    }
    else if(shape=='T')
    {
        printf("%d\n",side1*side2);
    }
    else
    {
        printf("0\n");
    }
}
```

	Input	Expected	Got	
✓	T 10 20	200	200	✓
✓	S 30 40	600	600	✓
✓	B 2 11	0	0	✓
✓	R 10 30	300	300	✓
✓	S 40 50	1000	1000	✓

Passed all tests! ✓

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 %)

```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d ",&n);
    n=n%296;
    int day=(n%10);
    switch(day)
    {
        case 0:
        printf("Sunday\n");
        break;
        case 1:
        printf("Monday\n");
        break;
        case 2:
        printf("Tuesday\n");
        break;
        case 3:
        printf("Wednesday\n");
        break;
        case 4:
        printf("Thursday\n");
        break;
        case 5:
        printf("Friday\n");
        break;
        case 6:
        printf("Saturday\n");
        break;
        case 7:
        printf("Kryptonday\n");
        break;
        case 8:
        printf("Coluday\n");
        break;
        case 9:
        printf("Daxamday\n");
        break;
    }
    return 0;

}
```

	Input	Expected	Got	
✓	7	Kryptonday	Kryptonday	✓
✓	1	Monday	Monday	✓

Passed all tests! ✓

Week 04-1

Output

Yes

Yes

No

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main ()
{
    int t;
    scanf("%d",&t);
    while (t--)
    {
        int n,r;
        scanf("%d",&n);
        r=n/4;
        if((r%2==0&&n%2==1)||(r%2==1&&n%2==0))
    {
        printf("Yes\n");
    }
    else
    {
        printf("No\n");
    }
    }
}
```

	Input	Expected	Got	
✓	3 1 6 7	Yes Yes No	Yes Yes No	✓

Passed all tests! ✓

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

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n,digit;
    scanf("%d",&n);
    int s=0;
    while (n!=0)
    {
        digit=n%10;
        if(digit==8)
        {
            s+=2;
        }
        else if ((digit==9)||(digit==0||digit==6))
        {
            s+=1;
        }
        n/=10;
    }
    printf("%d",s);
}
```

	Input	Expected	Got	
✓	630	2	2	✓
✓	1288	4	4	✓

Passed all tests! ✓

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

ut 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 a

nswer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int c=0;
    while (n>0)
    {
        c++;
        n/=2;
    }
    printf("%d",c);
}
```

	Input	Expected	Got	
✓	10	4	4	✓
✓	5	3	3	✓
✓	20	5	5	✓
✓	500	9	9	✓
✓	1000	10	10	✓

Passed all tests! ✓

Week 04-2

5

Explanation:

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

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n,x=0;
    while (scanf ("%d",&n)==1)
    {
        if (n%2!=0)
    {
        x++;
    }
    }
    printf("%d",x);
    }
```

	Input	Expected	Got	
✓	5 10 15 20 25 30 35 40 45 50	5	5	✓

Passed all tests! ✓

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 %)

```c
#include<stdio.h>
int main()
{
    int a, ch;
    scanf("%d",&a);
    while (a!=0)
    {
        int b=a%10;
        a=a/10;
        switch(b)
        {
            case 0:
            case 6:
            case 8:
            case 9:
            ch=0;
            break ;
            default:
            ch=1;
        }
    }
    if (ch==1)
    printf("false");
    else
    printf("true");
    return 0;
}
```

	Input	Expected	Got	
✓	6	true	true	✓
✓	89	true	true	✓
✓	25	false	false	✓

Passed all tests! ✓

? + 3 = 5, is the best case for maximum nutrients.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
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);
}
```

	Input	Expected	Got	
✓	2 2	3	3	✓
✓	2 1	2	2	✓
✓	3 3	5	5	✓

Passed all tests! ✓

Week 5-01

WBWBW

BWBWB

WBWBW

BWBWB

WBWBW

Answer: (penalty regime: 0 %)

```c
#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;
    }
    return 0;
}
```

	Input	Expected	Got	
✓	2 3 5	WBW BWB WBW WBWBW BWBWB WBWBW BWBWB WBWBW	WBW BWB WBW WBWBW BWBWB WBWBW BWBWB WBWBW	✓

Passed all tests! ✓

2 W

3 B

Output:

WB

BW

BWB

WBW

BWB

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main()
{
    int T,d,i,i1,i2,o,z;
    char c,s;
    scanf ("%d",&T);
    for(i=0;i<T;i++)
    {
        scanf("%d %c",&d,&s);
        for (i1=0;i1<d;i1++)
        {
            z=(s=='W')?0:1;
            o=(i1%2==z)?0:1;
            for(i2=0;i2<d;i2++)
            {
                c=(i2%2==o)?'W':'B';
                printf("%c",c);
            }
            printf("\n");
        }
    }
}
```

	Input	Expected	Got	
✓	2 2 W 3 B	WB BW BWB WBW BWB	WB BW BWB WBW BWB	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
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");
        }

    }
}
```

	Input	Expected	Got	
✓	3 3 4 5	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 102030405026027028029030 **607080902202302402 5 ****10011012019020021 ******13014017018 ********15016	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 102030405026027028029030 **607080902202302402 5 ****10011012019020021 ******13014017018 ********15016	✓

Passed all tests! ✓

Week 5-02

```c
#include<stdio.h>
#include<math.h>
int main ()
{
    int n;
    scanf("%d",&n);
    int x=0,n2=n;
    while(n2!=0)
    {
        x++;
        n2=n2/10;
    }
    int sum=0;
    int n3=n ,n4;
    while(n3!=0)
    {
        n4=n3%10;
        sum =sum+pow(n4,x);
        n3=n3/10;
    }
    if(n==sum)
    {
        printf("true");
    }
    else
    {
        printf("false");
    }
    return 0;
}
```

	Input	Expected	Got	
✓	153	true	true	✓
✓	123	false	false	✓

Passed all tests! ✓

```c
#include<stdio.h>
int main()
{
    int rn,n,nt=0,i=0;
    scanf("%d",&n);
    do
    {
        nt=n;rn=0;
        while(n!=0)
        {
            rn=rn*10+n%10;
            n=n/10;
        }
        n=nt+rn;
        i++;
    }
    while(rn!=nt||i==1);
    printf("%d",rn);
    return 0;
}
```

	Input	Expected	Got	
✓	32	55	55	✓
✓	789	66066	66066	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main ()
{
    int n=1,i=0,nt,co=0,e;
    scanf("%d",&e);
    while(i<e)
    {
        nt=n;
        while(nt!=0)
        {
            co=0;
            if(nt%10!=3 && nt%10!=4)
            {
                co=1;
                break;
            }
            nt=nt/10;
        }
        if(co==0)
        {
            i++;
        }
         n++;
    }
    printf("%d",--n);
}
```

	Input	Expected	Got	
✓	34	33344	33344	✓

Week 6-01

```c
#include<stdio.h>
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);
    }
}
```

	Input	Expected	Got	
✓	1 3 1 3 5 4	1	1	✓
✓	1 3 1 3 5 99	0	0	✓

Passed all tests! ✓

```c
#include<stdio.h>
int main()
{
    int t;
    scanf("%d",&t);
    while (t--)
    {
        int n,c=0;
        scanf("%d",&n);
        for (int i=0; i<=n;i++)
        {
            if (i%2!=0) c=c+i;
        }
        printf("%d\n",c);
    }
}
```

	Input	Expected	Got	
✓	3 1 2 3	1 1 4	1 1 4	✓
✓	10 71 100 86 54 40 9 77 9 13 98	1296 2500 1849 729 400 25 1521 25 49 2401	1296 2500 1849 729 400 25 1521 25 49 2401	✓

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
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);
    }
}
```

	Input	Expected	Got	
✓	4 1 4 2 4 2 3 5	2 4	2 4	✓
✓	5 2 10 5 4 8 4 3 1 7 8	1 0 3 4	1 0 3 4	✓

Passed all tests! ✓

Week 07

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
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]);
        }
        for(int a=0;a<n;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;
        }
    }
}
```

	Input	Expected	Got	
✓	2 4 5 1 4 5 3 2 4 4 2 2 4 3	1 4 1 2	1 4 1 2	✓

Passed all tests! ✓

```c
#include<stdio.h>
int main()
{
    int n,m,c,c1=0,c0;
    scanf("%d",&n);
    int arr[n];
    for(int a=0;a<n;a++)
    {
        scanf("%d",&arr[a]);
    }
    scanf("%d",&m);
    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++)
    {
        c0=0;
        for(int b=0;b<c1;b++)
        {
            if(ans[b]<ans[a])
            c0++;
        }
        int temp=ans[a];
        ans[a]=ans[c0];
        ans[c0]=temp;
    }
    for(int i=0;i<c1;i++)
    printf("%d ",ans[i]);
}
```

	Input	Expected	Got	
✓	10 203 204 205 206 207 208 203 204 205 206	204 205 206	204 205 206	✓

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 %)

```c
#include<stdio.h>
int main()
{
    int t,n,is,rs,m;
    scanf("%d",&t);
    for(int i=0;i<t;i++)
    {
        is=0;
        rs=0;
        scanf("%d",&n);
        int arr[n];
        for(int j=0;j<n;j++)
        scanf("%d",&arr[j]);
        m=n/2;
        if(arr[m]==0)
        for(m=0;arr[m]==0&&m<n;m++);
        for(int j=0;j<=m;j++)
        is=is+arr[j];
        for(int j=m;j<n;j++)
        rs=rs+arr[j];
        printf("%s\n",(is==rs)?"YES":"NO");
    }
}
```

	Input	Expected	Got	
✓	3 5 1 1 4 1 1 4 2 0 0 0 4 0 0 2 0	YES YES YES	YES YES YES	✓
✓	2 3 1 2 3 4 1 2 3 3	NO YES	NO YES	✓

Passed all tests! ✓

Week 08

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int t;
    scanf("%d",&t);
    while (t--)
    {
        int n,m,d,min,temp;
        scanf("%d %d",&n,&m);
        d=n-m;
        int arr[n];
        for(int i=0;i<n;i++)
        {
            scanf("%d",&arr[i]);
        }
        for(int j=0;j<n-1;j++)
        {
            min =j;
            for(int k=j;k<n;k++)
            {
                if(arr[k]<arr[min])
                min=k;
            }
            temp=arr[min];
            arr[min]=arr[j];
            arr[j]=temp;
        }
        int maxsum=0,minsum=0;
        for(int a=0;a<d;a++)
        {
            minsum+=arr[a];
        }
        for(int b=n-1;b>m-1;b--)
        maxsum+=arr[b];
        printf("%d\n",maxsum-minsum);
    }
}
```

	Input	Expected	Got	
✓	1 5 1 1 2 3 4 5	4	4	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n,min1,min2,temp,flag=1;
    scanf("%d",&n);
    int vac[n],pat[n];
    for(int i=0;i<n;i++)
    scanf("%d",&vac[i]);
    for(int i=0;i<n;i++)
    scanf("%d",&pat[i]);
    for (int j=0;j<n-1;j++)
    {
        min1=j;min2=j;
        for(int k=j;k<n;k++)
        {
            if(vac[k]<vac[min1])
            min1=k;
            if(pat[k]<pat[min2])
            min2=k;
        }
        temp=vac[min1];
        vac[min1]=vac[j];
        vac[j]=temp;
        temp=pat[min2];
        pat[min2]=pat[j];
        pat[j]=temp;
    }
    for(int i=0;i<n;i++)
    {
        if(vac[i]<=pat[i])
        {
            flag=0;
            break;
        }
    }
    if (flag==1)
    printf("Yes");
    else
    printf("No");
}
```

	Input	Expected	Got	
✓	5 123 146 454 542 456 100 328 248 689 200	No	No	✓

Passed all tests! ✓

ver: (penalty regime: 0 %)

```c
#include<stdio.h>
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;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            if((arr[i]^arr[j])==0)
            count++;
        }
    }
    printf("%d",count);
}
```

	Input	Expected	Got	
	5 1 3 1 4 3	2	2	✓

sed all tests! ✓

SAMPLE OUTPUT

```
4 2 0 1 3
```

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    scanf("%d",&arr[i]);
    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;
    }
}
```

	Input	Expected	Got	
✓	5 4 5 3 7 1	4 2 0 1 3	4 2 0 1 3	✓

Passed all tests! ✓

Week 09

1 2 3 4 5 6 7 8 9

SAMPLE OUTPUT

25

20

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main ()
{
    int arr[3][3];
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            scanf("%d",&arr [i][j]);
        }
    }
    int odd=0,even=0;
    for(int i=0; i<3;i++)
    {
        for(int j=0; j<3;j++)
        {
            if((i+j)%2!=0)
            odd+=arr[i][j];
            else
            even+=arr[i][j];
        }
    }
    printf("%d\n%d",even,odd);
}
```

	Input	Expected	Got	
✓	1 2 3 4 5 6 7 8 9	25 20	25 20	✓
✓	21 422 423 443 586 645 657 846 904	2591 2356	2591 2356	✓

Passed all tests! ✓

```c
struct data
{
    int gen; int tal;

};
int main ()
{
    int n;
    scanf("%d",&n);
    struct data a[n];
    for( int i=0;i<n;i++)
    scanf("%d %d",&a[i].gen,&a[i].tal);
    for(int i=0;i<n-1;++i)
    {
        for(int j=0;j<n-i-1;++j)
          {
              if(a[j].tal <a[j+1].tal)
              {
                struct data temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
              }
          }
    }
    for (int i=0;i<n;i++)
    {
       if(a[i].gen==0)
       printf("%d ",a[i].tal);
    }
    for (int i=0;i<n;++i)
    {
        if(a[i].gen==1)
        printf("%d ",a[i].tal);
    }
}
```

	Input	Expected	Got	
✓	5 0 3 1 6 0 2 0 7 1 15	7 3 2 15 6	7 3 2 15 6	✓
✓	6 0 1 0 26 0 39 0 37 0 7 0 13	39 37 26 13 7 1	39 37 26 13 7 1	✓
✓	12 1 12 1 14 1 18 1 1	31 29 18 14 12 10 9 8 5 3 2 1	31 29 18 14 12 10 9 8 5 3 2 1	✓

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main ()
{
    int i,j,n,x1,x2,y1,y2,t=0;
    long long total=0;
    int arr[1001][1001]={0};
    scanf("%d",&n);
    while(n--)
    {
        scanf("%d  %d %d %d %d",&x1,&y1,&x2,&y2,&t);
        for(i=x1;i<=x2;i++)
        {
            for (j=y1;j<=y2;j++)
            {
                if(arr[i][j]==0)
                arr[i][j]+=t;
                else if(arr[i][j]>0)
                arr[i][j]=(-1)*(arr[i][j]+t);
                else if (arr[i][j]<0)
                arr [i][j]-=t;
            }
        }
    }
    for(i=1;i<1001;i++)
    {
        for(j=1;j<1001;j++)
        {
            if(arr[i][j]<0)
            total+=arr[i][j];
        }
    }
    printf("%lld\n",(-1)*total);
    return 0;
}
```

	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! ✓

Week 10

the given string:

1 occurs two times.

2, 4, 5, 6 and ***7*** occur one time each.

he remaining digits ***0, 3, 8*** and ***9*** don't occur at all.

nswer: (penalty regime: 0 %)

```c
#include<stdio.h>
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]);
    }
}
```

	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	✓
✓	1v888861256338ar0ekk	1 1 1 2 0 1 2 0 5 0	1 1 1 2 0 1 2 0 5 0	✓

Passed all tests! ✓

In test case 1, a and o are the only vowels. So, count=2

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main ()
{
    int t;
    scanf("%d",&t);
    while (t--)
    {
        char str[100000];
        int count =0;
        scanf ("%s",str);
        for (int i=0;str[i]!= '\0';i++)
        {
            char c=str[i];
            if((c=='a')||(c=='e')||(c=='i')||(c=='o')||(c=='u')||(c=='A')||(c=='E')||(c=='I')||(c=='O')||(c=='U'))
            count ++;
        }
        printf("%d\n",count);
    }
}
```

	Input	Expected	Got	
✓	2 nBBZLaosnm JHkIsnZtTL	2 1	2 1	✓
✓	2 nBBZLaosnm JHkIsnZtTL	2 1	2 1	✓

Passed all tests! ✓

This

is

C

Explanation 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 %)

```c
#include <stdio.h>
int main()
{
    char s[10000];
    scanf("%[^\n]s",s);
    for (int i=0; s[i]!='\0';i++)
    {
        if (s[i]!=' ')
        printf("%c",s[i]);
        else
        printf("\n");
    }
}
```

	Input	Expected	Got	
✓	This is C	This is C	This is C	✓
✓	Learning C is fun	Learning C is fun	Learning C is fun	✓

Passed all tests! ✓

Explanation

a = "abcd"

b = "ef"

|a| = 4

|b| = 2

a + b = "abcdef"

a' = "ebcd"

b' = "af"

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main ()
{
    char str1[10],str2[10],t;
    int i=0,j=0;
    int count1=0,count2=0;
    scanf("%s",str1);
    scanf("%s",str2);
    while ( str1[i]!='\0')
    {
        count1++;
        i++;
    }
    while (str2[j]!='\0')
    {
        count2++;
        j++;
    }
    printf("%d %d\n",count1,count2);
    printf("%s%s\n",str1,str2);
    t=str1[0];
    str1[0]=str2[0];
    str2[0]=t;
    printf("%s %s",str1,str2);
}
```

	Input	Expected	Got	
✓	abcd ef	4 2 abcdef ebcd af	4 2 abcdef ebcd af	✓

Week 11

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

ver: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<string.h>
int main()
{
    char str1[1000000],str2[1000000];
    int flag = 1;
    scanf("%s",str1);
    scanf("%s",str2);
    int a = strlen(str1);
    int b = strlen(str2);
    if(a==b)
    {
        for(int i=a-1;i>=0;i--)
        {
            while(str1[i]!=str2[i])
            {
                for(int j=0;j<=i;j++)
                {
                    if(str1[j]<'z')
                    str1[j]++;
                    else
                    {
                        flag = 0;
                        break;
                    }
                    if(flag==0)
                     break;
                }
            }
        }
    }
    else
    flag=0;
    if(flag==0)
    printf("NO");
    else
    printf("YES");
}
```

	Input	Expected	Got	
	abaca cdbda	YES	YES	✓

sed all tests! ✓

swer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<string.h>
int main()
{
    int n,flag=0;
    char temp;
    scanf("%d",&n);
    char words [n] [n];
    for(int i=0;i<n;i++)
    scanf("%s",words[i]);
    char reverse[14];
    for(int i=0;i<n;i++)
    {
        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;
}
```

	Input	Expected	Got	
✓	4 abc def feg cba	3 b	3 b	✓

assed all tests! ✓

Dominos

Explanation

Dominos has maximum points.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<string.h>
int main()
{
    int n;
    scanf ("%d",&n);
    char res[n][21];
    int rate[n];
    for(int i=0;i<n;i++)
    {
        scanf("%s",res[i]);
        scanf("%d",&rate[i]);
    }
    int max=rate[0];
    char ans[20];
    strcpy(ans,res[0]);
    for(int i=1;i<n;i++)
    {
        if(rate[i]>max)
        {
            max=rate[i];
            strcpy(ans,res[i]);
        }
        else if(rate[i]==max)
        {
            if(strcmp(res[i],ans)<0)
           strcpy(ans,res[i]);
        }
    }
    printf("%s",ans);
}
```

	Input	Expected	Got	
✓	3 Pizzeria 108 Dominos 145 Pizzapizza 49	Dominos	Dominos	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<string.h>
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        int flag=1;
        char s[100000];
        scanf("%s",s);
        int k =strlen(s);
        if(k==10)
        {
            for(int i=0;i<10;i++)
            {
                if(s[0]=='0')
                {
                    flag=0;
                    break;
                }
                if(s[i]<'0'||s[i]>'9')
                {
                    flag=0;
                    break;
                }
            }
        }
        else
        flag=0;
        if(flag==1)
        printf("YES\n");
        else
        printf("NO\n");
    }
}
```

	Input	Expected	Got	
✓	3 1234567890 0123456789 0123456.87	YES NO NO	YES NO NO	✓

Passed all tests! ✓

Week 12

Answer: (penalty regime: 0 %)

Reset answer

```c
/*
 * Complete the 'myFunc' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER n as parameter.
 */

int myFunc(int n)
{
    return n==1 || n%10==0;
}

```

	Test	Expected	Got	
✓	printf("%d", myFunc(1))	1	1	✓
✓	printf("%d", myFunc(2))	0	0	✓
✓	printf("%d", myFunc(10))	1	1	✓
✓	printf("%d", myFunc(25))	0	0	✓
✓	printf("%d", myFunc(200))	1	1	✓

Passed all tests! ✓

mple Output 2

planation 2

0 can be expressed as the sum of the cubes of *1, 2, 3, 4*.

+ 8 + 27 + 64 = 100). There is no other way to express *100* as the sum of cubes.

swer: (penalty regime: 0 %)

Reset answer

```c
/*
 * Complete the 'powerSum' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts following parameters:
 *  1. INTEGER x
 *  2. INTEGER n
 */
#include<math.h>
int powerSum(int x, int m, int n)
{
    int  p=pow(m,n);
    if(p==x)
    {
        return 1;
    }
    if(p>x)
    {
        return 0;
    }
    return powerSum(x-p,m+1,n) + powerSum(x,m+1,n);
}
```

	Test	Expected	Got	
✓	printf("%d", powerSum(10, 1, 2))	1	1	✓

assed all tests! ✓

Week 11

Reset answer

```c
/*
 * Complete the 'fourthBit' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER number as parameter.
 */

int fourthBit (int number)
{
    int binary[32];
    int i=0;
    while(number>0)
    {
        binary[i]=number%2;
        number/=2;
        i++;

    }
    if(i>=4)
    {
        return binary[3];
    }
    else
    return 0;
}

```

	Test	Expected	Got	
✓	printf("%d", fourthBit(32))	0	0	✓
✓	printf("%d", fourthBit(77))	1	1	✓

Passed all tests! ✓

Reset answer

```c
/*
 * Complete the 'pthFactor' function below.
 *
 * The function is expected to return a LONG_INTEGER.
 * The function accepts following parameters:
 *  1. LONG_INTEGER n
 *  2. LONG_INTEGER p
 */

long pthFactor(long n, long p)
{
    int count=0;
    for(long i=1;i<=n;i++)
    {
        if(n%i==0)
        {
            count++;
            if(count==p)
            {
                return i;
            }
        }
    }
    return 0;
}
```

	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! ✓

Reset answer

```c
/*
 * Complete the 'pthFactor' function below.
 *
 * The function is expected to return a LONG_INTEGER.
 * The function accepts following parameters:
 *  1. LONG_INTEGER n
 *  2. LONG_INTEGER p
 */

long pthFactor(long n, long p)
{
    int count=0;
    for(long i=1;i<=n;i++)
    {
        if(n%i==0)
        {
            count++;
            if(count==p)
            {
                return i;
            }
        }
    }
    return 0;
}
```

	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! ✓

Week 14

nswer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        int length,width,height;
        scanf("%d %d %d",&length,&width,&height);
        if(height <41)
        {
            int volume=length*width*height;
            printf("%d\n",volume);
        }
    }
}
```

	Input	Expected	Got	
✓	4 5 5 5 1 2 40 10 5 41 7 2 42	125 80	125 80	✓

Passed all tests! ✓

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
typedef struct
{
    double area;
    int a,b,c;
}
Triangle;
double calculate_area(int a,int b,int c)
{
    double p=(a+b+c)/2.0;
    return sqrt(p*(p-a)*(p-b)*(p-c));
}
int compare(const void*x,const void*y)
{
    Triangle*t1=(Triangle*)x;
    Triangle*t2=(Triangle*)y;
    if(t1->area < t2->area) return -1;
    if(t1->area > t2->area) return 1;
    return 0;
}
int main()
{
    int n;
    scanf("%d",&n);
    Triangle triangles[n];
    for(int i=0;i<n;i++)
    {
        int a,b,c;
        scanf("%d %d %d",&a,&b,&c);
        triangles[i].a=a;
        triangles[i].b=b;
        triangles[i].c=c;
        triangles[i].area=calculate_area(a,b,c);

    }
    qsort(triangles,n,sizeof(Triangle),compare);
    for(int i=0;i<n;i++)
    {
        printf("%d %d %d\n",triangles[i].a,triangles[i].b,triangles[i].c);
    }
    return 0;
}
```

	Input	Expected	Got	
✓	3 7 24 25 5 12 13 3 4 5	3 4 5 5 12 13 7 24 25	3 4 5 5 12 13 7 24 25	✓

Week 15

Reset answer

```c
/*
 * Complete the 'reverseArray' function below.
 *
 * The function is expected to return an INTEGER_ARRAY.
 * The function accepts INTEGER_ARRAY arr as parameter.
 */

/*
 * To return the integer array from the function, you should:
 *     - Store the size of the array to be returned in the result_count variable
 *     - Allocate the array statically or dynamically
 *
 * For example,
 * int* return_integer_array_using_static_allocation(int* result_count) {
 *     *result_count = 5;
 *
 *     static int a[5] = {1, 2, 3, 4, 5};
 *
 *     return a;
 * }
 *
 * int* return_integer_array_using_dynamic_allocation(int* result_count) {
 *     *result_count = 5;
 *
 *     int *a = malloc(5 * sizeof(int));
 *
 *     for (int i = 0; i < 5; i++) {
 *         *(a + i) = i + 1;
 *     }
 *
 *     return a;
 * }
 *
 */
int* reverseArray(int arr_count, int *arr, int *result_count)
{
    *result_count = arr_count;
    for(int i=0;i<arr_count/2;i++)
    {
        int temp=arr[i];
        arr[i]=arr[arr_count-i-1];
        arr[arr_count-i-1]=temp;
    }
    return arr;
}
```

	Test	Expected	Got	

✓	int arr[] = {1, 3, 2, 4, 5};
	int result_count;
	int* result = reverseArray(5, arr, &result_count);
	for (int i = 0; i < result_count; i++)
	printf("%d\n", *(result + i));
		5
		4
		2
		3
		1
			5
			4
			2
			3
			1
				✓

Passed all tests! ✓

```c
/*
 * Complete the 'cutThemAll' function below.
 *
 * The function is expected to return a STRING.
 * The function accepts following parameters:
 *  1. LONG_INTEGER_ARRAY lengths
 *  2. LONG_INTEGER minLength
 */

/*
 * To return the string from the function, you should either do static allocation or dynamic allocation
 *
 * For example,
 * char* return_string_using_static_allocation() {
 *     static char s[] = "static allocation of string";
 *
 *     return s;
 * }
 *
 * char* return_string_using_dynamic_allocation() {
 *     char* s = malloc(100 * sizeof(char));
 *
 *     s = "dynamic allocation of string";
 *
 *     return s;
 * }
 *
 */
char* cutThemAll(int lengths_count, long *lengths, long minLength)
{
    long t=0,i=1;
    for(int i=0;i<=lengths_count-1;i++)
    {
        t+=lengths[i];
    }
    do
    {
    if(t-lengths[lengths_count-i-1]<minLength)
    {
        return "Impossible";

    }
    i++;
}
while(i<lengths_count-1);
return "Possible";
}
```

	Test	Expected	Got	
✓	long lengths[] = {3, 5, 4, 3}; printf("%s", cutThemAll(4, lengths, 9))	Possible	Possible	✓
✓	long lengths[] = {5, 6, 2}; printf("%s", cutThemAll(3, lengths, 12))	Impossible	Impossible	✓

Passed all tests! ✓

Week 13

Answer: (penalty regime: 0 %)

Reset answer

```c
/*
 * Complete the 'balancedSum' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY arr as parameter.
 */

int balancedSum(int arr_count, int* arr)
{
    int totalsum = 0;
    for(int i=0;i<arr_count;i++)
    {
        totalsum+=arr[i];

    }
    int leftsum=0;
    for(int i=0;i<arr_count;i++)
    {
        int rightsum=totalsum-leftsum-arr[i];
        if(leftsum==rightsum)
        {
            return i;
        }
        leftsum+=arr[i];
    }
    return 1;
}

```

	Test	Expected	Got	
✓	`int arr[] = {1,2,3,3};` `printf("%d", balancedSum(4, arr))`	2	2	✓

Passed all tests! ✓

Calculate the sum of an array of integers.

12 + 12 = 24.

Answer: (penalty regime: 0 %)

Reset answer

```c
/*
 * Complete the 'arraySum' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY numbers as parameter.
 */

int arraySum(int numbers_count, int *numbers)
{
    int sum=0;
    for(int i=0;i<numbers_count;i++)
    {
        sum=sum+numbers[i];
    }
    return sum;
}

```

	Test	Expected	Got	
✓	`int arr[] = {1,2,3,4,5};` `printf("%d", arraySum(5, arr))`	15	15	✓

Passed all tests! ✓

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 Case 0 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. Sample Case 1 Sample Input For Custom Testing STDIN Function ----- -------- 2 → arr[] size n = 2 3 → arr[] = [3, 2] 2 Sample Output 1 Explanation n = 2 arr = [3, 2] There is no need to rearrange because there are only two elements. The final answer is |3 - 2| = 1.

Answer: (penalty regime: 0 %)

Reset answer

```c
/*
 * Complete the 'minDiff' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY arr as parameter.
 */
#include<stdio.h>
int compare(const void*a,const void*b)
{
    return (*(int*)a - *(int*)b);
}
int minDiff(int arr_count, int* arr)
{
    qsort(arr, arr_count,sizeof(int),compare);
    int totaldiff=0;
    for(int i=1;i<arr_count;i++)
    {
        totaldiff+=abs(arr[i]-arr[i-1]);
    }
    return totaldiff;
}
```

	Test	Expected	Got	
✓	`int arr[] = {5, 1, 3, 7, 3};` `printf("%d", minDiff(5, arr))`	6	6	✓

Passed all tests! ✓

Finish review