ZOHO

Test Summary

- No. of Sections: 3
- No. of Questions: 171
- Total Duration: 400 min

Section 1 - ZOHO 2

Section Summary

- No. of Questions: 40
- Duration: 200 min

Additional Instructions:

None

Q1. ***Remove palindrome words***
string should contains only the words are not palindrome

Input Format

Input is a string

Output Format

Print the altered string

Sample Input

```
Malayalam is my mother tongue
```

Sample Output

```
is my mother tongue
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. **Frequency Sorting**
Given an array of integers arrange them in the descending order of their frequencies

Sample Input

```
11
1 2 3 1 1 1 3 2 4 4 2
```

Sample Output

```
1 1 1 1 2 2 2 3 3 4 4
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3. 10 + 20 = _____

Q4. ***First maximum and first minimum and so on***
Given an array and arrange it with first maximum and first minimum and second maximum and second minimum and so on without using sorting and second array

Input Format

Input will have size and the values

Output Format

Print the required output

Constraints

1<=size<=1000

Sample Input

```
15
5 15 10 25 55 35 75 45 95 50 70 40 60 90 3
```

Sample Output

```
95 3 90 5 75 10 70 15 60 25 55 35 50 40 45
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5. **BINARY - DECIMAL**
A positive integer is called Binary-Decimal it contains only 0's and 1's
Sample Input0:
32
Sample Output0:
11 11 10
3
Explanation:
There are many possibilities for representing 32 as a sum of Binary-Decimals
Few possibilities will be
10 + 10 + 1+ 1
Count = 5
11 + 10 + 10 + 1

Count = 4
11+11+10
Count = 3
The Expected output is(11 + 11 + 10) as it has minimum number of Binary-Decimals
(Count – 3)

Sample Input

```
32
```

Sample Output

```
11 11 10
3
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q6. **SNAKE PATTERN**
(DONT use MATRIX)

Input Format

N = 5

Output Format

```
        1  2  3  4  5
       10  9  8  7  6
      11 12 13 14 15
    20 19 18 17 16
21 22 23 24 25
```

Sample Input

```
5
```

Sample Output

```
        1  2  3  4  5
       10  9  8  7  6
      11 12 13 14 15
    20 19 18 17 16
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q7. **SPLIT THE STRING ACCORDING TO DICTIONARY**
Given an input string and a dictionary of word,find out if the input string can be segmented into a space separated sequence of dictionary words

Consider the following dictionary
{i , like , ice , cream , icecream }
the input string is ilikeicecream the expected output is
i like icecream

Input Format

N - no of words in dictionary
dictionary of words
input string

Output Format

display the separated input string

Sample Input

```
5
i like ice cream icecream
ilikeicecream
```

Sample Output

```
i like icecream
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q8. **LATIN SQUARE**
write a program to construct a latin square of a given N without using
i) any conditional statements (if - else / ternary operator)
ii) matrix
A latin square is an n X n matrix array filled with n different symbols , each occurring exactly once in each row and exactly once in each column

Sample Input

```
3
```

Sample Output

```
A B C

C A B
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q9. **PATTERN**

Sample Input

```
5
```

Sample Output

```
1
1 1
2 1
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q10. **DECOMPRESS THE STRING**

Assume that the given string has enough memory
Don't use any extra space(IN-PLACE)

Sample Input

```
a2b4c6
```

Sample Output

```
aabbbbccccccc
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q11. ***Convert number to words***
range is 0-999

Sample Input

```
234
```

Sample Output

```
two hundred and thirty four
```

Sample Input

```
200
```

Sample Output

```
two hundred only
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q12. ***Spiral pattern***

Sample Input

```
5
```

Sample Output

```
5 5 5 5 5 5 5 5 5
5 4 4 4 4 4 4 4 5
5 4 3 3 3 3 3 4 5
5 4 3 2 2 2 3 4 5
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q13. ***Alternate sort in unsorted array***
(no extra space)

Sample Input

```
9
23 7 8 30 18 12 6 28 16
```

Sample Output

```
23 7 18 12 16 28 8 30 6
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q14. ***Print the string***
note : no extra space
works only on odd length string
if the string is welcome

```
w              e
   e         m
      l   o
         c
      l   o
   e         m
w              e
```

Input Format

Input will be a string

Output Format

Print the string in the given format

Sample Input

```
welcome
```

Sample Output

```
w              e
   e         m
      l   o
         c
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q15. ***Find a Sub string***
Find if a String2 is substring of String1. If it is, return the index of the first occurrence. else return -1.

Sample Input

```
thistest123string
123
```

Sample Output

```
8
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q16. ***Merge two sorted arrays without duplication***
Output is a merged array without duplicates

Input Format

N1 - no of elements in array 1
array elements for array 1
N2 - no of elements in array 2
array elements for array2

Output Format

display the merged array

Sample Input

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

Sample Output

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

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q17. ***Remove unbalanced parenthesis***
Remove unbalanced parentheses in a given expression.

Sample Input

```
((abc)((de))
```

Sample Output

```
(abc)((de))
```

Sample Input

```
(((ab)
```

Sample Output

```
(ab)
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q18. ***Array with threshold value***
Given an array and a threshold value find the o/p
i/p {5,8,10,13,6,2}
threshold = 3
o/p 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

```
6
5 8 10 13 6 2
3
```

Sample Output

```
17
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q19. ***Triangle Pattern***
Note : Don't use any matrix
N= 7

```
1 8 14 19 23 26 28
2 9 15 20 24 27
3 10 16 21 25
4 11 17 22
5 12 18
6 13
7
```

Sample Input

```
5
```

Sample Output

```
1 6 10 13 15
2 7 11 14
3 8 12
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q20. ***Find the different pair***
print the pair which are mismatched in two strings

Input Format

Input will have two strings

Output Format

Print the mismatched pair separated by comma

Sample Input

```
abcdefgh
abdfhjfb
```

Sample Output

```
c,d d,f e,h f,j g,f h,b
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q21. Sliding Window
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

```
8
1 3 5 2 1 8 6 9
3
```

Sample Output

```
5 5 5 8 8 9
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q22. In the following line print only the words that are not palindrome
Sample Input0: He did a good deed
Sample Output0: He good
Sample Input1: Malayalam is my mother tongue
Sample Output1: is my mother tongue

Input Format

Input contains the string

Output Format

Print the altered string

Constraints

Should not use extra memory
1 <= length <= 1000

Sample Input

```
He did a good deed
```

Sample Output

```
He good
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q23. ***Sorting Based on no of factors***
To find the factors of the numbers given in an array and to sort the numbers in descending order according to the factors present in it

Input Format

Input contains the array size and the values

Output Format

print the array which is sorted by the factors count

Constraints

1 <= array_size <= 1000

Sample Input

```
5
8 2 3 12 16
```

Sample Output

```
12 16 8 2 3
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q24. ***Odd numbers in a range***
To find the odd numbers in between the range.

Input Format

Input represents two integers start and end range

Output Format

Print the odd numbers separated by space

Constraints

1<= start<=end<=1000000

Sample Input

```
2 15
```

Sample Output

```
3 5 7 9 11 13
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q25. ***Evaluate mathematical expression***
Check whether a given mathematical expression is valid.

Sample Input

```
(a+b)(c+d+e)
```

Sample Output

```
VALID
```

Sample Input

```
(a+b)(c+d)
```

Sample Output

```
VALID
```

Sample Input

```
(a+b))
```

Sample Output

```
INVALID
```

Sample Input

```
(ab+)
```

Sample Output

```
INVALID
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q26. ***Leap Year***
find whether the given year is leap year or not

Input Format

Input will be an integer

Output Format

Print Leap / Non-leap

Sample Input

```
1990
```

Sample Output

```
Non-leap
```

Sample Input

```
2000
```

Sample Output

```
Leap
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q27. ***Number and its occurrence***
Given a array with n elements print the number of occurrences of that number each number in that array. The order of number doesn't matter. You can reorder the elements.

Input Format

N - no of elements
array of elements

Output Format

display number followed by counts

Sample Input

```
10
4 7 18 16 14 16 7 13 10 2
```

Sample Output

```
4-1
7-2
18-1
16-2
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q28. ***Reverse a String***

Given a string as input, you have to reverse the string by keeping the punctuation and spaces intact. You have to modify the source string itself without creating another string.

Constraints

1<=string length<=500

Sample Input

```
A man, in the boat says : I see 1-2-3 in the sky
```

Sample Output

```
y kse, ht ni3 21ee sIsy : a sta o-b-e ht nin amA
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q29. ***String pattern***

print the string as the following pattern
(only for odd length string)

Sample Input

```
Hello
```

Sample Output

```
  l
 ll
llo
lloH
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q30. ***WORD REVERSAL USING RECURSIVE***

Using Recursion to reverse the string such as

Input Format

one two three

Output Format

three two one

Constraints

1 <= string length <= 200

Sample Input

```
i love india
```

Sample Output

```
india love i
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q31. ***cyclic number verification***

verify the given number is cyclic or not

Input Format

Num1 num2

Constraints

1<=range<=9999999999

Sample Input

```
12345 45123
```

Sample Output

```
Yes
```

Sample Input

```
12345 54123
```

Sample Output

```
No
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q32. ***Pattern***

Sample Input

Sample Output

4	4444 4334 4334 4444

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q33. ***Reverse and Add until get a palindrome***
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 **Sample Output**

32	55

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q34. ***Remove Characters***
Given two Strings s1 and s2, remove all the characters from s1 which is present in s2.

Input Format

Input will have two strings

Output Format

Print the string

Constraints

1<= string length <= 200

Sample Input **Sample Output**

experience enc	xpri

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q35. **Excel Sheet**
Given a number, convert it into corresponding alphabet.
Input Output
1 A
26 Z
27 AA
676 YZ

Input Format

Input is an integer

Output Format

Print the alphabets

Constraints

1 <= num <= 4294967295

Sample Input **Sample Output**

26	Z

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q36. **Roman to Decimal**
Given a Roman numeral, find its corresponding decimal value.

Input Format

Input is a string which contains Roman numbers

Output Format

Print the decimal value

Constraints

1<=string_length<100

Sample Input **Sample Output**

XLV	45

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q37. ***New Number System***
Form a number system with only 3 and 4. Find the nth number of the number system.
Eg.) The numbers are: 3, 4, 33, 34, 43, 44, 333, 334, 343, 344, 433, 434, 443, 444, 3333, 3334, 3343, 3344, 3433, 3434, 3443, 3444 ….

Input Format

Input will be an integer

Output Format

Print the nth number

Constraints

1<=N<=10000000

Sample Input

```
10
```

Sample Output

```
344
```

Sample Input

```
6743
```

Sample Output

```
434334344333
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q38. ***Add 2 numbers in given Base***
Add two numbers in the given base without converting into base

Input Format

get two numbers and base

Output Format

display the sum

Sample Input

```
1010 11001 2
```

Sample Output

```
100011
```

Sample Input

```
123  13 4
```

Sample Output

```
202
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q39. **Excel Sheet**
Given a number, convert it into corresponding alphabet.
Input Output
1 A
26 Z
27 AA
676 YZ

Input Format

Input is an integer

Output Format

Print the alphabets

Constraints

```
1 <= num <= 4294967295
```

Sample Input

```
26
```

Sample Output

```
Z
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q40. **Check if a number is a power of another number**
Given two positive numbers x and y, check if x is a power of y or not.

Examples :

Input: x = 100, y = 5
Output: True

Input: x = 1000, y = 10

Output: True

Input: x = 10001, y = 10
Output: False

Sample Input

```
625 5
```

Sample Output

```
True
```

Sample Input

```
128 5
```

Sample Output

```
False
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Section 2 - ZOHO 1

Section Summary

- No. of Questions: 125
- Duration: 100 min

Additional Instructions:

None

Q1. 1. Find the output for the following programs

```
#include<stdio.h>
Void main()
{
int i;
for( i = 1 ; i < 4 ; i++)
{
 switch(i)
 {
   case 1 : printf("%d" , i);break;
   case 2 : printf("%d" , i);break;
   case 3 : printf("%d" , i);break;
 }

}
switch(i)
{
   case 4 : printf("%d" , i);break;
}
} _____
```

Q2. 1. Find the output

```
void main()
{
   int a[10] , i = 0;
   for(i = 0 ; i<10 ; i++)
        a[i] = 9 - i;
   for(i = 0 ; i < 10 ; i++)
        a[i] = a[a[i]];
   for(i = 0 ; i < 10 ; i++)
        printf("%d " , a[i]);
} _____
```

Q3. 1. Find the output

```
void printout(char* pstr)
{
   int iretval = 0;
   if(pstr)
   {
      while(*pstr && *pstr<= '9' && *pstr >= '0')
      {
         iretval = (iretval*10)+(*pstr - '0');
         pstr++;
      }
   }
 printf("%d\n" , iretval);
}
void main()
{
   printout("X32");
   printout("47X74");
} _____
```

Q4. 1. Find the output

```
void main()
{
char *s = "\12345s\n";
```

```
printf("%d" , sizeof(s));
} _____
```

Q5.

1. Find the output

```
void main()
{
  int a[] = { 8 , 9 , 9 , 9};
  int r[5] = { 0 , 0 , 0 , 0 , 0};
  int i = 0 , m = 1 , s = 4;
  for( i = s-1 ; i >= 0 ; i--)
  {
     r[i+1] = (a[i] + m ) % 10;
     m = (a[i] + m) /10;

  }
  r[0] = m ;
  for( i = 1 ; i <= s ; i++)
      printf("%d" , r[i]);
} _____
```

Q6.

1. Find the output

```
int i = 0 ;
int fun(int a)
{
  i++;
  if( a > 99)
      return a-12;
  return fun(fun(a+25));
}
void main()
{
  printf("%d " , fun(69));
  printf("%d" , i);
} _____
```

Q7.

1. Find the output ons)

```
int max(int x , int y)
{
  return (y > x) ? y : x;
}
void main()
{
int a[] = { -6 , - 7 , 8 , - 9 , -2 , 3,-4,5};
int value = a[0] , ctval = a[0];
int i = 0 , n = 8;
for( i = 1 ; i < n ; i++)
{
  ctval = max( a[i] , ctval+a[i]);
  value = max(value , ctval);
}
 printf("%d" ,value) ;
} _____
```

Q8.

1. Find the output

```
void main()
{
  int a=30,b=40,x;
  x=(a=10)&(b=50);
  printf("x=%d\n",x);
} _____
```

Q9.

1. Find the outputint main()

```
{
char *s[]={ "dharmr'a","hewlett-packard","siemens","ibm"};
char **p;
p = s ;

printf("%s" ,++*p);
printf("%s",*p++); ;
printf("%s" ,++*p);
} _____
```

Q10.

1. Find the output

```
void main()
{
  int x=0,y=1;
  y=x;
  x=!y;
  printf("x=%d y=%d\n",x,y);
} _____
```

Q11.

1. Find the output

```
void main()
{
  int x=3,y=4,z=4;
```

```
  printf("ans=%d\n",(z>=y>=x?100:200));
} _____
```

Q12.

1. Find the output

```
void main()
{
  float a=12.25,b=13.65;
  if(a==b)
      printf("a and b are equal");
  else
     printf("a and b are not equal");
} _____
```

Q13.

1. Find the output

```
void main()
{
int arr[]={0,1,2,3,4};
int *ptr,i;
for(ptr=arr+4;ptr >= arr ; ptr--)
 printf("%d ",*ptr);
} _____
```

Q14.

Find the output

```
void junk(int,int*);
int main()
{
int i=-5,j=-2;
junk(i,&j);
printf("i=%d j=%d" ,i ,j);

return 0;
}
void junk(int i,int       *j)
{
 i=i*1;
*j=*j * i;
} _____
```

Q15.

Find the output

```
void main()
{
  if('Z' < 'z')
     printf("Pilots are on strike...\n");
  else
     printf("for absolutely outlandish demands\n");

} _____
```

Q16.

Find the output

```
void main()
{
  float a = 0.7;
  if( a < 0.7)
      printf("Stoned");
  else
     printf("Avenged");
}
_____
```

Q17.

Find the output

```
void main()
{
  float a=0.5,b=0.9;
  if(a&&b>0.9)
     printf("tce-cse-a\n");
  else
     printf("tce-cse-b\n");
} _____
```

Q18.

1. Find the output(branching and looping)

```
void main()
{
  int i;
  for(i=1;i++<=5;printf("%d ",i));
}
_____
```

Q19.

Find the output

```
void main()
{
  int i = 1 , j = 1;
  for(;j;printf("%d %d\n",i,j))
    j=i++<=5;
```

```
}
```
_____

Q20. Find the outputvoid main()

```
{
  int x=3,y,z;
  z=y=x;
  z*=y=x*x;
  printf("x=%d y=%d z=%d\n",x,y,z);
}
```
_____

Q21. Find the output

```
void main()
{
  int x=3,z;
  z=x/++x;
  printf("x=%d z=%d\n",x,z);

}
```
_____

Q22. Find the output

```
void main()
{
  int x , y , z;
  x=y=z=1;
  z=++x || ++y&&++z;
  printf("x=%d y=%d z=%d \n",x,y,z);

}
```
_____

Q23. Find the output

```
int main()
{
char *s[]={ "dharmr'a","hewlett-packard","siemens","ibm"};
char **p;
p = s ;

printf("%s" ,++*p);
printf("%s",*p++); ;
printf("%s" ,++*p);
}
```
Output: _____

Q24. Find the output

```
void main()
{
  char ch='E';
  switch(ch)
  {
     case(ch>=65 && ch<=90):
        printf("Capital letter\n");
        break;
     case(ch>=97 && ch<=122):
        printf("small letter\n");
        break;
     case (ch>=48 && ch<=57):
        printf("Digit");
        break;
     default:printf("Anyother");
  }
```
_____

Q25. Find the output

```
void main()
{
  int i = 3;
  switch(i)
  {
     case 1: printf("cse\t");
     case 2: printf("It\n");break;
     case 3: continue;
     default : printf("goodbye");
  }
}
```
_____

Q26. Find the output

```
void main()
{
  char s;
  switch(s)
  {
     case '1': printf("database");
```

```
    case '2': printf("data-structure");
    default: printf("c");
    printf("byebye");
  }
}
```

Q27.

```
Find the output
void main()
{
  int k=-2,j=4;
  switch(k/=j/k)
  {
     default:printf("lenovo");
     case 0 : printf("hp");
     case 1: printf("acer");
     case 2: printf("dell");
  }
}
```

Q28.

```
Find the output
void main()
{
  int j,x=0;
  for (j=0;j<=5;j++)
  {
     switch(j-1)
     {
     case 0:
     case -1:
        x -= 1;break;
     case 1:
     case 2:
     case 3:
         break;
     default: x+=3;

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

}
```

Q29.

```
Find the output
void main()
{
  int i;
  for(i = 2 ; i <= 10 ; i++)
  {
     switch(i)
     {
        case 2: printf("0");continue;
        case 3: break;
        case 4:
        case 5:printf("1");break;
        default: printf("000");
     }
  }
}
```

Q30.

```
Find the output
void main()
{
  char ch='E';
  switch(ch) {
  case(ch>=65 &&ch<=90):
     printf("Capital letter\n");
     break;
   case(ch>=97 &&ch<=122):
      printf("small letter\n");
      break;
   case (ch>=48&&ch<=57):
      printf("Digit");
      break;
   default: printf("Any other");

  }

}
```

Q31.

```
Find the output
#include<stdio.h>
#include<malloc.h>
#include<string.h>
int main()
{
int i;
```

```
char a[]="String";
char *p = "New String";
char *temp;
temp = malloc(strlen(p) + 1);
p = malloc( strlen(temp) + 1);
strcpy(p , temp);
printf("%s" , p);
}
```
Output : _____

Q32.
```
Find the output
int funcl(int k)
{
  k++;
  return k;
}

void main()
{
  int k=35,z;
  k=funcl(k=funcl(k=funcl(k)));
  printf("k =%d\n",k);

}
```
_____

Q33.
```
Find the output
int main()
{
int n = 12 , res = 1;
while( n > 3)
{
  n -= 3;
  res *= 3;
}
printf("%d" , n*res);
}
```
Output : _____

Q34.
```
Find the output
void fun(int [][3]);
int main()
{
int a[3][3] = {9,8,7,6,5,4,3,2,1};
fun(a);
printf("%d\n" , a[2][1]);
}
void fun(int b[][3])
{
  ++b;
  b[1][1]=5;
}
```
Output : _____

Q35.
```
Find the outputvoid pri(int,int);
void printit(float,int);
void main()
{
 float a=3.14;
int i=99;
pri(i,a);
printit(a,i);


}
void pri(int i,int a)
{
printf("i=%d a=%f\n",i,a);
printf("a=%f i=%d\n",a,i);
}
void printit(float a,int i)
{
printf("a=%f i=%d\n",a,i);
printf("i=%d a=%f\n",i,a);
}
```
_____
_____
_____
_____

Q36.
```
Find the output
void main()
{

  int i , n;
  char x[5];
  strcpy( x , "Zoho");
  n = strlen(x);
  *x = *(x+(n-1));
  printf("%s" , x);
}
```

Output: _____

Q37. Find the output

```
void main()
{
  int c[]={5,4,3,4,5};
  int j , *q = c;
  for( j = 0 ; j<5 ; j++){
      printf("%d" , *c);
      ++q;
  }
}
```

Output: _____

Q38. Find the output

```
void main(){
int k=35,*z,*y;
z=&k; y=z;
*z++=*y++;
k++;
printf("k=%d z=%d y=%d",k,z,y);
}
```

_____

Q39. Find the output

```
void main()
{
  int i = 1;
  for(i =0 ; i= -1 ; i=1){
      printf("%d", i);
  if(i!= 1) break;
  }
}
```

Output: _____

Q40. Find the output

```
void main()
{

  int s[] = {1,0,5,0,10,0};
  int f[] = {2,4,6,8,10,12};
  int n = 6 , i = 0 , j = 0;
  for( j = 1 ; j < n ; j++)
  {
     if( s[j] >= f[i])
     {
        printf("%d" , i);
        i = j;
     }
  }
}
```

output : _____

Q41. Find the output

```
void main(){
int a=100,*b,**c,***d;
b=&a; c=&b; d=&c;
printf("%d %d %d %d",a,*b,**c,***d);

}
```

_____

Q42. Find the output

```
void main(){
int z=4;
printf("%d\n",printf("%d%d\n",z, z-1));
}
```

_____

Q43. Find the output

```
void f(int *a , int m)
{
  int j = 0;
  for(j = 0 ; j < m ; j++)
  {
     *(a+j) = *(a+j) - 5;
  }
}
void main()
{
  int a[] = {'f' , 'g' , 'h' , 'i' , 'j'};
  int j = 0 ;
  f(a , 5);
  for(j = 0 ; j<= 4 ; j++)
     printf("%c\t" , a[j]);
```

```
}
```
Output: _____

Q44. Find the output

```
float *jam(float *r){

r=r+1;
return (r);
}


void main()
{
float *jam(float *);
float p=23.5,*q;
q=&p;
printf("q before call=%d\n",q);
q=jam(&p);
printf("q after call=%d",q);
}
```
_____
_____

Q45. Find the output

```
void main()
{
int i;
printf("hai");
for(i = 1 ; i<= 10 ; i++)
   main();

}
```
_____

Q46. Find the output

```
void main()
{
  int i=0, j=0 , sum=0;
  for(i= 1; i < 500 ; i*=3)
    for(j=0;j<i;j++)
      sum++;
  printf("%d",sum);

}
```
Output: _____

Q47. Find the output

```
void main()
{
if(printf("C for yourself how it works\n"))
 main();

}
```
_____

Q48. Find the output

```
void main()
{
  int n;
  for(n = 6 ; n!= 1; n--)
      printf("%d" , n--);
}
```
Output: _____

Q49. Find the output

```
void main()
{
unsigned int ch=0;

for(ch=65;ch<=255;)
  printf("%d %c\n",ch,ch++);

}
```
_____

Q50. Find the output

```
void main()
{
  int a[3][4] = {2,4,6,5,10,12,12,10,5,6,4,2};
  int i = 0 , j , k =99;
  while(i < 3)
  {
     for(j = 0 ; j < 4 ; j= j++)
     {
        if( a[i][j] < k)
        {
```

```
            k = a[i][j];
          }
      }
      i++;
  }
  printf("%d" , k);
}
```
Output : _____

Q51.
```
Find the output
void main()
{
float a=0.7;
double b=0.7;
if(a==b)
  printf("condition statisfied");
else
  printf("condition not statisfied");

printf("\na=%f b=%lf\n",a,b);
}
```
_____
_____

Q52.
```
Find the output
void main()
{

char *x="Alice";
int i , n = strlen(x);
*x = x[n];
for(i=0; i<=n; i++)
{
printf("%s ", x); x++;
printf("\n", x);
}

return 0;


}
```
Output : _____

Q53.
```
Find the output
void main()
{
float y=0.9;
long double z=0.9;

if(y=-z)
 printf("icecrearm");
else
  printf("cake");

}
```
_____

Q54.
```
Find the output
void change()
{
  auto int i=100;
        register int j=200;
printf("change's i and j are %d %d\n",i,j);

}
void main()
{
auto int i=10;
register int j=20;
printf("main's I and j are %d %d\n",i,j);
change();
printf("main's I and j are %d %d\n",i,j);
}
```
_____
_____
_____

Q55.
```
Find the output
void main()
{
double x,d=4.4;
int i=2,y;
x=(y=d/i)*2;
printf("x=%lf y=%d\n",x,y);
y=(x=d/i)*2;
printf("x=%lf y=%d\n",x,y);
}
```
_____
_____

Q56. Find the output

```
void main()
{
double x,d=5.0;
int y;
x=d*(x=2.5/d);
printf("x=%lf\n",x);
x=d*(y=(int)2.5+1.5);
printf("x=%lf y=%d\n",x,y);
}
```

_____

_____

Q57. Find the output

```
struct value{
int bit1:1;
int bit3:4;
int bit4:4;
}bit;
int main()
{
printf("%d\n", sizeof(bit));
return 0;
}
```

Output : _____

Q58. Find the output

```
void main()
{
double x,d=5.0;
int y;
x=d*(x=2.5/d);
printf("x=%lf\n",x);
x=d*(y=(int)2.5+1.5);
printf("x=%lf y=%d\n",x,y);
}
```

_____

_____

Q59. Find the output

```
struct node
{
int data;
float d;
struct node *link;
};
int main()
{
struct node *p, *q;
p = (struct node *) malloc(sizeof(struct node));
q = (struct node *) malloc(sizeof(struct node));
printf("%d, %d\n", sizeof(p), sizeof(q));
return 0;
}
```

Output : _____

Q60. Find the output

```
void main()
{
int c=5;
printf("c=%d\n",c--);
if(c) main();
}
```

_____

Q61. Find the output

```
int func(int x)
{
  static int v=2;
  v--;
  return (v-x);
}
int i;
void main()
{
int j;
for(;;)
{
  if( j= func(i) )
   printf("j= %d " , j);
  else
     break;
}

}
```

_____

Q62. Find the output

```
void main()
{
long num=2;
short n=2;
signed no=2;
printf("num=%ld n=%d no=%d\n",num,n,no);
}
```

_____

Q63. Find the output

```
void main()
{
char ch=122,ch1='z';
printf("ch=%c\n",ch);
printf("chl=%d\n",ch1);
}
```

_____

_____

Q64. Find the output

```
typedef union
{
  int a;
  char b[10];
  float c;
}Union;
int main()
{
  Union x , y = {100};
  x.a = 50;
  strcpy(x.b , "Hello");
  x.c = 21.50;
  printf("%d %s %f\n" , x.a , x.b , x.c);
  printf("%d %s %f" , y.a,y.b, y.c);

}
```

Output: _____

_____

Q65. Find the output

```
void main()
{
unsigned int a=25;
unsigned b=25;
long unsigned c=345L;
long signed d=345L;
printf("a=%u b=%u\n",a,b);
printf("c=%lu d=%d\n",c,d);
}
```

_____

Q66. Find the output

```
struct point{

int x;
int y ;
};
struct point origin , *pp;
int main()
{
  pp = &origin;
  printf("origin is (%d %d)\n", (*pp).x , (*pp).y);
  printf("origin is (%d %d)" , pp->x , pp->y);
  return 0;
}
```

Output : _____

_____

Q67. Find the output

```
void main()
{
int i = -1;
printf("i =%d +i = %d\n" , i , +1);
}
```

Output : _____

Q68. Find the output

```
void main()
{
char not;
not=12;
printf("%d",not);

}
```

Output : _____

Q69. Find the output

```
void main()
{
auto int i=100;
printf("i=%d\n",i);
i+=1;
printf("i=%d\n",i);

}
```

_____

_____

Q70. Find the output

```
#p1
void main()
{
register int i;
for(i=1;i<=100;i++)
 printf("%d\n",i);

}
#p2
void main()
{
auto int i;
for(i=1;i<=100;i++)
 printf("%d\n",i);

}
```

_____

Q71. Find the output

```
#define FALSE -1
#define TRUE 1
#define NULL 0
void main()
{
if(NULL)
 puts("NULL");
else if(FALSE)
 puts("TRUE");
else
 puts(" FALSE");

}
```

Output : _____

Q72. Find the output

```
#define CUBE(x) x*x*x
void main()
{
int a;
a= 27 / CUBE(3);
printf("%d" , a);

}
```

_____

Q73. Find the output

```
#define CUBE(x) (x*x*x)
void main()
{
int a , b;
a = CUBE(b+4) / b++;
printf("a= %d b = %d ", a , b);

}
```

_____

Q74. Find the output

```
void main()
{
  int k = 1;
  printf("%d==1 is"" %s" ,k, k == 1 ? "TRUE":"FALSE");
}
```

Output : _____

Q75. Find the output

```
#define AND &&
#define OR ||
#define LE <=
#define GE >=
void main()
{
char ch='D';
if((ch GE 65 AND ch LE 90) OR (ch GE 97 AND ch LE 122))
  printf("Alphabet\n");
```

```
else
   printf("Not alnhabet");

}
```
_____

Q76. Find the output

```
void main()
{
static float arr[]={1.2,12,2.4,24,3.5,35};
int i;
for(i=0;i<=5;i++)
 printf("%f ",arr[i]);
}
```
_____

Q77. Find the output

```
void main()
{
static int b[]={10,20,30,40,50};
int i;
for(i = 0; i<= 4 ; i++)
 printf("%d ",b[i]);
}
```
_____

Q78. Find the output

```
int main()
{
int t , i ;
for ( t=4;scanf("%d",&i)-t;printf("%d\n",i))
   printf("%d--",t--);

}
```
Output : _____

Q79. Find the output

```
void main()
{
static int a[5]={5,10,15,20,25};
int i,j,m,n;
i=4-a[1];
j=a[1]++;
printf("i=%d j=%d a[1]=%d\n",i,j,a[1]);
i=1;
m=a[1]+41;
printf("i =64 m=%d\n",i,m);
i=2;
n=a[1]++;
printf("i=%d n=%d\n",i,n);

}
```
_____
_____
_____

Q80. Find the output

```
struct emp{
int len;
char name[1];
};
int main()
{
char newname[] = "Rahul";
struct emp *p = (struct emp *) malloc(sizeof(struct emp) -1 + strlen(newname)+ 1);
p->len = strlen(newname);
strcpy(p -> name, newname);
printf("%d %s\n", p->len, p->name); return 0;
}
```
Output : _____

Q81. Find the output

```
void main()
{
static int a[]={10,20,30,40,50};
int j;
for (j=0;j<5;j++)
{
printf("%cl\n",*a);
a++;
}
```
_____

Q82. Find the output

```
int main() {
printf("%d %d %d %d\n",72,072,0x72,0X72);
```

```
return 0;
}
```
Output : _____

Q83. Find the output

```
void main()
{
 char ch;
 int a;
 float b;
printf("bytes occupied by ch=%d\n",sizeof(ch));
printf("bytes occupied by a=%d\n",sizeof(a));
printf("bytes occupied by b=%d\n",sizeof(b));
}
```
Output :

_____

_____

_____

Q84. Find the output

```
void main()
{
   printf("%d\n" , sizeof('7'));
   printf("%d\n" , sizeof(7));
   printf("%d\n" , sizeof(7.0));
}
```
Output: _____

_____

_____

Q85. Find the output

```
void main()
{
   char ch=291;
   printf("%d %d %c\n",2147483648,ch,ch);
   return 0;
}
```
Output : _____

Q86. Find the output

```
void main()
{
static int b[]={10,20,30,40,50};
int i,*k;
k= &b[4]-4;
for(i=0;i<=4;i++)
printf("%d ",*k);
k++;

}
```
_____

Q87. Find the output

```
void main()
{
   int g;
   g=300000*300000/300000;
   printf("g=%d\n",g);
}
```
Output : _____

Q88. Find the output

```
void main()
{
static int a[]={2,4,6,8,10};
int i;
for(i=0;i<=4;i++)
 *(a+i)=a[i]+i[a];
printf("%d\n",*(i+a));


}
```

_____

Q89. Find the output

```
void main()
{
   float a;
a=4/2;
printf("%f %f\n",a,4/2);

}
```
Output : _____

Q90. Find the outputvoid main()

```
{
int arr[]={0,1,2,3,4};
int i,*ptr;
for(ptr=&arr[0],i=0;i<=4;i++)
  printf("%d ",ptr[i]);
}
```

_____

Q91. Find the output

```
void main()
{
  printf("%d\n",sizeof(4)/sizeof(2.0));
  printf("%d\n",sizeof(2.0)/sizeof(4));
}
```

Output : _____

Q92. Find the output

```
void main()
{
int arr[]={0,1,2,3,4};
int i,*p;
for(p=arr,i=0;p+i<=arr+4;p++,i++)
  printf("%d ",*(p+i));

}
```

_____

Q93. Find the output

```
void main()
{
  int x=10,y=5,p,q;
  p=x > 9;
  q=x>3&& y!=3;
  printf("p=%d q=%d \n",p,q);

}
```

Output : _____

Q94. Find the output

```
void main()
{
static int a[]={0,1,2,3,4};
static int *p[]={a,a+1,a+2,a+3,a+4};
int **ptr=p;
printf("%d %d\n",a,*a);
printf("%d %d %d\n",p,*p,**p);
printf("%d %d %d\n",ptr,*ptr,**ptr);
}
```

_____
_____
_____

Q95. Find the output

```
void main()
{
static int a[]={0,1,2,3,4};
static int *p[]={a,a+1,a+2,a+3,a+4};
int **ptr=p;
printf("%d %d %d\n",ptr-p,*ptr-a,**ptr);
*ptr++;
printf("%d %d %d\n",ptr-p,*ptr-a,**ptr);
*++ptr;
printf("%d %d %d\n",ptr-p,*ptr-a,**ptr);
++*ptr;
printf("%d %d %d\n",ptr-p,*ptr-a,**ptr);
}
```

_____
_____
_____
_____

Q96. Find the output

```
void main()
{
static int n[3][3]={12,4,3,6,8,5,3,5,11};
printf("%d %d %d\n",n,n[2],n[2][2]);
}
```

_____

Q97. Find the output

```
void main()
{
char s[]="Rendezvous !";
printf("%d\n",*(s+strlen(s)));
```

```
}
```
_____

Q98. Find the output

```
void main()
{
char str[20];
static int i;
for(;;) {
  i++[str]='A'+2;
  if(i==19)
    break;
}
i[str]=0;
printf("%s" , str);

}
```
_____

Q99. 1. Find the output(strings)

```
void main()
{
char s[]="C smart!!";
int i;
for(i=0;s[i];i++)
 printf("%c%c%c%c\n",s[i],*(s+i),i[s],*(i+s));

}
```

Output :

_____

Q100. 1. Find the outputvoid main()

```
{
char s[]="Dinks Grunts and Guffaws";
printf("%c\n",*(&s[2]));
printf("%s\n",(s+5));
printf("%s\n",s);
printf("%c\n",*(s+10));
}
```
_____
_____
_____
_____

Q101. Find the output

```
void main()
{
  int a = 3 , b = 2 , c = 1 , d;
  d = a | b & c;
  printf("d = %d\n", d);
  d = a+ b & -c;
  printf("d = %d\n" , d);
}
```
_____

Q102. Find the output

```
void main()
{
char str[]="MalayalaM";
char *s;
s = str+8;
while( s > str)
{
   printf("%c" , *s);
   s--;
}
}
```
_____

Q103. Find the output

```
int show();
void main()
{

  int (*f)();
  f= show;
  printf("address= %d\n",f);
}
int show()
{
  printf("Diamonds are very costly");
}
```
_____

Q104. Find the output

```
void main()
```

```
{
char str[]="Shall we tell the Deputy Director?";
printf("%s\n%s\n%s\n",str,str+6,str+9);

}
```

Q105. Find the output

```
struct num
{
  unsigned bit0:1;
  unsigned bit1:1;
  unsigned bit2:1;
  unsigned rest:5;
};
union a
{
  struct num n;
  char ch;
}b;
void main()
{
b.ch = 32;
printf("%d %d %d %d", b.n.bit0 , b.n.bit1,b.n.bit2,b.n.rest);
}
```

Q106. Find the output

```
void main()
{
printf("%f\n" , (float)(int)(float)(int)6.5/2+3.5);
}
```

Q107. Find the output

```
struct employee
{
  char name[25];
  int age;
  float bs;
};

void main()
{
struct employee e;
e.name = "Hacker";
e.age=25;
printf("%s %d" , e.name , e.age);

}
```

Q108. Find the output

```
typedef struct
{
  char name[20];
  int age;
}a;
void main()
{
 a temp= {"sunil" , 30};
 printf("%s %d" , temp.name , temp.age);

}
```

Q109. Find the output

```
struct name1
{
  char name[25];
  char lang[10];
};
static struct name1 a = {"Hacker" , "cr"};
void main()
{
printf("%s %s" , a.name , a.lang);
}
```

Q110. Find the output

```
void main()
{
 enum status {low,medium,high};
 enum status rain;
 rain = 0;
 if(rain == low)
```

```
    printf("rain = %d", rain);
}
```
_____

Q111. Find the output

```
struct a {
  char ch[7];
  char *str;

};

void main()
{
static struct a s1={"Nagpur" , "Bombay"};
printf("%c %c\n" , s1.ch[0] , *s1.str);
printf("%s %s" , s1.ch , s1.str);
}
```
_____
_____

Q112. Find the output

```
void main()
{
 int a=0Xff;
 if(a<<4>>12)
   printf("leftest");
 else
   printf("rightest");
}
```
_____

Q113. Find the output

```
void main()
{
 int a=0Xff;
 if(a<<4>>12)
   printf("leftest");
 else
   printf("rightest");
}
```
_____

Q114. Find the output

```
struct a
{
  int i;
  char ch[4];
};
union b
{
  int j;
  char ch[4];
};
void main()
{
printf("%d " , sizeof(struct a));
printf("%d " , sizeof(union b));
}
```
_____

Q115. Find the output

```
void main()
{
  short int k;
  k = -35;
  printf("k=%d " , k);
  k = -k;
  printf("k = %d " , k);
}
```
_____

Q116. Find the output

```
int main(int argc , char* argv[])
{
  printf("%d ", argc);
  printf("%s" , argv[0]);
}
```
_____

Q117. Find the output

```
union a
{
 int i;
 char ch[2];
};
void main()
```

```
{
union a u;
u.i = 256;
printf("%d %d %d " , u.i , u.ch[0] , u.ch[1]);
}
```
_____

Q118. Find the output

```
{
  long int i;
  char ch[4];
};
void main()
{
struct a s;
s.i = 512;
printf("%d %d %d" , s.ch[0] , s.ch[1] ,s.ch[3]);
}
```
_____

Q119. Find the output

```
void main()
{
static char str[]="Triplet";
char *s;
s = str;
while(*s)
{
  putc(*s , stdout);
  fputchar(*s);
  printf("%c\n ",*s);
  s++;
}
}
```
_____

Q120. Find the output

```
void main()
{
char name[20]="Sandeep";
int salary=1500;
printf("%s %d\n", name , salary);
fprintf(stdout , "%s%d\n",name,salary);
}
```
_____
_____

Q121. Find the output

```
{
  int i;
  char ch[4];
};
void main()
{
union a u;
u.ch[0]=3;
u.ch[1]=2;
u.ch[2]=0;
u.ch[3]=0;
printf("%d %d %d",u.ch[0],u.ch[1], u.i);
}
```
_____

Q122. Find the output

```
void main()
{
printf("I\tam\ta\tboy\n");
}
```
_____

Q123. Find the output

```
void main()
{
float a=3.14;
printf("a=%f\n",a);
printf("a=%6.2f\n",a);
printf("a=%-6.2\n",a);
printf("a=%6.1f\n",a);
printf("a=%6.0f\n",a);
}
```
_____
_____
_____
_____
_____

Q124. Find the output

```
void main()
{
printf("Hello\nHi\n");
printf("Hello\rHi\n");
printf("Hello\b\b\b\b\b\n");
printf("Hil\b\b\bBye\n");
}
```

_____
_____
_____
_____
_____

Q125. Find the output

```
void main()
{
printf("%20\s\n","short leg");
printf("%20\s\n","long leg");
printf("%20\s\n","deep fine leg");
printf("%20\s\n","backward short leg");
printf("%20\s","legs are the same");
}
```

_____
_____
_____
_____
_____

Section 3 - ZOHO

Section Summary

- No. of Questions: 6
- Duration: 100 min

Additional Instructions:

None

Q1. ***New Number System***

Form a number system with only 3 and 4. Find the nth number of the number system.
Eg.) The numbers are: 3, 4, 33, 34, 43, 44, 333, 334, 343, 344, 433, 434, 443, 444, 3333, 3334, 3343, 3344, 3433, 3434, 3443, 3444 ….

Input Format

Input will be an integer

Output Format

Print the nth number

Constraints

1<=N<=10000000

Sample Input

```
10
```

Sample Output

```
344
```

Sample Input

```
6743
```

Sample Output

```
434334344333
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. ***Word Reversal form the first occurrence of sub-string***

Write a program to accept two strings S1 and S2 and reverse the words of S1,starting from the word where the first occurrence of S2 present in S1. Same empty spaces between the words must be maintained in the output. Write the program without splitting up the strings into array of words.
DON'T use any inbuilt functions
Input: S1= This is a test input string S2=st
Output : This is a string input test

Sample Input

```
this is a test sentence
st
```

Sample Output

```
this is a sentence test
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3. ***Permutation of string***

A permutation, also called an “arrangement number” or “order,” is a rearrangement of the elements of an ordered list S into a one-to-one correspondence with S itself. A string of length n has n! permutation.
Below are the permutations of string ABC.
ABC ACB BAC BCA CBA CAB

Sample Input

```
ABC
```

Sample Output

```
ABC ACB BAC BCA CBA CAB
```

Sample Input

```
1234
```

Sample Output

```
1234 1243 1324 1342 1432 1423 2134 2143 2314 2341 2431 2413 3214 324
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4. ***Locker Problem***

There is a school with 100 students, and correspondingly 100 lockers, all of which start off closed. The first student opens every locker. The second student closes every other locker, starting with the second (2, 4, 6 etc). The third student changes the state of every third locker starting with the third (3,6,9 etc). The fourth would change the status of lockers numbered 4,8,12 etc.,. That is, if the locker is open, it is closed, and if it is closed, it is opened. This continues until all 100 students have passed along the lockers. After the 100^{th} student is done, which lockers are open and which are closed?

[Note: program should work for any number of students/lockers]

Sample Input

```
100
```

Sample Output

```
open = 10
close = 90
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5. ***Find the Path***

Given an (m x n) matrix, write a program to traverse the cell and print the values present in the given path. Inclued necessary validation and proper error messages in case of given path is out of bounds.
5 x 5 matrix :
{1 2 3 4 5 } (row 1)
{6 7 8 9 0 } (row 2)
{1 2 3 4 5 } (row 3)
{6 7 8 9 0 } (row 4)
{1 2 3 4 5 } (row 5)

Path Notation : ”>” is going right, “v” going down, “<” is going left, “^” is going up.

Example Input 1 :
Start at (Row, Column): 1, 2
Path: >>> v
Output: 2 3 4 5 0

Example Input 2:
Start at (Row, Column): 2,3
Path: v > > v < < ^ > > v v

Output 8 3 4 5 0 9 8 3 4 5 0 5

Example Input 3:
Start at(row, Column): 1 , 4
Path: > v > >
Output: Invalid Path

Input Format

N M - matrix row and col
input for matrix
startrow startcol
path string

Sample Input

```
5 5
1 2 3 4 5
6 7 8 9 0
1 2 3 4 5
```

Sample Output

```
8 3 4 5 0 9 8 3 4 5 0 5
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q6. ***Group anagram words***

Given array of words, group the anagrams and print. Any word or phrase that exactly reproduces the letters in another order is an anagram. Arrive most efficient algorithm.

Examples :
Input: {tar,rat,banana,atr,nanaba}
Output: Anagrams:
rat atr tar
nanaba banana

Input: {abc, cde, xyz, dec}
Output: Anagrams:
cde dec
Others:

Abc
xyz

Input Format

N - no of words
get input words into array

Sample Input

```
5
tar rat banana atr nanaba
```

Sample Output

```
rat atr tar
nanaba banana
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Answer Key & Solution

Section 1 - ZOHO 2

Q1

Test Case

Input

```
He did a good deed
```

Output

```
He good
```

Weightage - 10

Input

```
Hah lovevol you
```

Output

```
you
```

Weightage - 15

Input

```
hjds abcba abcd ekdjcb
```

Output

```
hjds abcd ekdjcb
```

Weightage - 10

Input

```
mind Blowolb
```

Output

```
mind
```

Weightage - 15

Input

```
sdhfdh sdhdges dhfesjhfgesj
```

Output

```
sdhfdh sdhdges dhfesjhfgesj
```

Weightage - 10

Input

```
wonder of youoy
```

Output

```
wonder of
```

Weightage - 5

Input

```
sdhdshgdvcdhcvdhcsjc
```

Output

```
sdhdshgdvcdhcvdhcsjc
```

Weightage - 5

Input

```
aaaaabbbbaaaaa hsdcjkdcbjdskcbkjbd
```

Output

```
hsdcjkdcbjdskcbkjbd
```

Weightage - 10

Input

Output

zdvcdhcvdbvjdbvjzdvcdhcvdbvjdbvj cdc fdvhjvbdcvbhdsv	zdvcdhcvdbvjdbvjzdvcdhcvdbvjdbvj fdvhjvbdcvbhdsv

Weightage - 20

Sample Input

```
Malayalam is my mother tongue
```

Sample Output

```
is my mother tongue
```

Solution

```
#include<stdio.h>
#include<stdlib.h>
int mystrlen(char *str)
{
    int ind;
    for(ind = 0 ; str[ind] ; ind++);
return ind;
}
int isPalindrome(char *str)
{
    int start , end;
    for(start = 0 , end = mystrlen(str) - 1 ; start < end ; start++ , end--)
    {
        if(abs(str[start] - str[end]) == 32 )
             continue;
        if( str[start] != str[end] )
             return 0;
    }
return 1;
}
int main()
{
char str[100];
char *start;
int ind;
scanf("%[^\n]s" , str);
start = str;
for(ind = 0 ; str[ind] ; ind++)
{
    if( str[ind] ==  32)
    {
        str[ind] = 0;
        if( isPalindrome(start) == 0)
             printf("%s " , start);
        str[ind] = 32;
        start = str + ind + 1;
    }
}
if( isPalindrome(start) == 0)
             printf("%s " , start);
return 0;
}
```

Q2

Test Case

Input

```
6
-1 -1 4 5 2 2
```

Output

```
-1 -1 2 2 4 5
```

Weightage - 15

Input

```
12
4 4 2 1 3 4   1 3 4 5 6 1
```

Output

```
4 4 4 4 1 1 1 3 3 2 5 6
```

Weightage - 20

Input

```
15
1  2  3  4  5  6  2  3  5  2  5  6  3  2  3
```

Output

```
2  2  2  2  3  3  3  3  5  5  5  6  6  1  4
```

Weightage - 10

Input

```
5
1  2  3  4  5
```

Output

```
1  2  3  4  5
```

Weightage - 5

Input

```
12
1  2  3  2  23  123  33  23  87  23  83  9
```

Output

```
23  23  23  2  2  1  3  123  33  87  83  9
```

Weightage - 15

Input

```
5
12345  12345  24  43  3
```

Output

```
12345  12345  24  43  3
```

Weightage - 10

Input

```
10
10  39  349  27  9  10  9  7  9  10
```

Output

```
10  10  10  9  9  9  39  349  27  7
```

Weightage - 15

Input

```
5
3  3  2  5  3
```

Output

```
3  3  3  2  5
```

Weightage - 10

Sample Input

```
11
1  2  3  1  1  1  3  2  4  4  2
```

Sample Output

```
1  1  1  1  2  2  2  3  3  4  4
```

Solution

```c
#include <stdio.h>
#include<limits.h>
#include<malloc.h>
#define SIZE sizeof(arr) / sizeof(arr[0])

int main()
{
    int n , ind;
    scanf("%d" , &n);
    int arr[n] ;
    int** occurence = NULL;
    int maxpos, count = 0, count_flag, o_row, index,max;
    int newindex = 0, num, ctr;

    for(ind = 0 ; ind < n ; ind++)
         scanf("%d" , &arr[ind]);
    occurence = (int**)calloc(1,sizeof(int*));
```

```
occurence[0] = (int*)calloc(2,sizeof(int));
occurence[0][0] = arr[0];
occurence[0][1]++;
count++;
for(index = 1 ; index < SIZE ; index++)
{
        //search in occurence array
        for(o_row = 0, count_flag = 0 ;  o_row < count ; o_row++)
        {
                if(occurence[o_row][0]  == arr[index])
                {
                        occurence[o_row][1]++;
                        count_flag = 1;
                }
        } //search completed
         if(count_flag  == 0)
         {
                 occurence = (int**)realloc(occurence,(count+1)*sizeof(int*));
                 occurence[count] = (int*)calloc(2,sizeof(int));
                 occurence[count][0] = arr[index];
                 occurence[count][1]++;
                 count++;
         }
}
for(index = 0 ; index < count ; index++)
 {
                //find maximum in occurence
                for(o_row = 0, max = INT_MIN ; o_row < count ; o_row++)
                {
                        if(occurence[o_row][1]  >  max)
                        {
                                max = occurence[o_row][1];
                                maxpos = o_row;
                        }
                }
    num = occurence[maxpos][0];

    for(ctr = 1 ; ctr <= max ; ctr++)
             arr[newindex++] = num;
    occurence[maxpos][1] =  -1;
     }
     //printf("%d %d\n",max, maxpos);
    // for(index = 0 ; index < count ; index++)
    //printf("%d ,%d\n", occurence[index][0], occurence[index][1]);
    for(index = 0 ; index < SIZE ; index++)
          printf("%d ",arr[index]);
    for(index=0 ; index < count ; index++)
          free(occurence[index]);
    free(occurence);
    return 0;
}
```

```
#include <stdio.h>
#include<limits.h>
#include<malloc.h>
#define SIZE sizeof(arr) / sizeof(arr[0])

int main()
{
    int n , ind;
    scanf("%d" , &n);
    int arr[n] ;
    int** occurence = NULL;
    int maxpos, count = 0, count_flag, o_row, index,max;
    int newindex = 0, num, ctr;

    for(ind = 0 ; ind < n ; ind++)
         scanf("%d" , &arr[ind]);
    occurence = (int**)calloc(1,sizeof(int*));
    occurence[0] = (int*)calloc(2,sizeof(int));
    occurence[0][0] = arr[0];
    occurence[0][1]++;
    count++;
    for(index = 1 ; index < SIZE ; index++)
    {
            //search in occurence array
            for(o_row = 0, count_flag = 0 ;  o_row < count ; o_row++)
            {
                    if(occurence[o_row][0]  == arr[index])
```

```
                {
                        occurence[o_row][1]++;
                        count_flag = 1;
                }
        } //search completed
         if(count_flag  == 0)
         {
                occurence = (int**)realloc(occurence,(count+1)*sizeof(int*));
                occurence[count] = (int*)calloc(2,sizeof(int));
                occurence[count][0] = arr[index];
                occurence[count][1]++;
                count++;
         }
}
for(index = 0 ; index < count ; index++)
 {
               //find maximum in occurence
           for(o_row = 0, max = INT_MIN ; o_row < count ; o_row++)
           {
                  if(occurence[o_row][1]  >  max)
                  {
                         max = occurence[o_row][1];
                         maxpos = o_row;
                  }
           }
num = occurence[maxpos][0];
      for(ctr = 1 ; ctr <= max ; ctr++)
              arr[newindex++] = num;
      occurence[maxpos][1] =  -1;
       }
       //printf("%d %d\n",max, maxpos);
      // for(index = 0 ; index < count ; index++)
      //printf("%d ,%d\n", occurence[index][0], occurence[index][1]);
      for(index = 0 ; index < SIZE ; index++)
            printf("%d ",arr[index]);
      for(index=0 ; index < count ; index++)
            free(occurence[index]);
      free(occurence);
      return 0;
  }
```

Q3 30

Solution

Q4

Test Case

Input

```
7
7 9 3 6 8 2 1
```

Output

```
9 1 8 2 7 3 6
```

Weightage - 5

Input

```
7
467 742 8 32 8 489 26
```

Output

```
742 8 489 8 467 26 32
```

Weightage - 15

Input

```
8
200 4 84 82 62 8 38 643
```

Output

```
643 4 200 8 84 38 82 62
```

Weightage - 15

Input

Output

```
7
486 73 8942 2742 831913 947 1
```

```
831913 1 8942 73 2742 486 947
```

Weightage - 15

Input

Output

```
8
47 743 5 853 835 735 53 88
```

```
853 5 835 47 743 53 735 88
```

Weightage - 15

Input

Output

```
13
44 556 235 56 546 6564 45 35 2 678 4367 75 7783
```

```
7783 2 6564 35 4367 44 678 45 556 56 546 75 235
```

Weightage - 10

Input

Output

```
10
45 454 437 7 98 74 6724 89 674 7
```

```
6724 7 674 7 454 45 437 74 98 89
```

Weightage - 15

Input

Output

```
7
378 75 85 85 623 9 89
```

```
623 9 378 75 89 85 85
```

Weightage - 10

Sample Input

Sample Output

```
15
5 15 10 25 55 35 75 45 95 50 70 40 60 90 3
```

```
95 3 90 5 75 10 70 15 60 25 55 35 50 40 45
```

Solution

```
#include<stdio.h>
#include<malloc.h>
void swap(int* , int,int);
void maxHeapify(int* , int* , int);
void minHeapify(int* , int* , int);
int main()
{
int ind , parent,left,right , n ;
//int arr[]={5,15,10,25,55,35,75,45,95,50,70,40,60,90,3};
int *arr ;
scanf("%d" , &n);
arr = (int*)malloc( n * sizeof(int));
for(ind = 0 ; ind < n ; ind++)
    scanf("%d" ,&arr[ind]);

int safe , count = 0;
safe = n ;
while(n!=0)
{
for(parent = n / 2 - 1;parent >=0 ; parent--)
{
    if(count % 2 == 0)
       maxHeapify(arr , &n , parent);
    else
       minHeapify(arr , &n , parent);
}
swap(arr , 0, --n);

count++;
```

```
#include<stdio.h>
#include<malloc.h>
void swap(int* , int,int);
void maxHeapify(int* , int* , int);
void minHeapify(int* , int* , int);
int main()
{
int ind , parent,left,right , n ;
//int arr[]={5,15,10,25,55,35,75,45,95,50,70,40,60,90,3};
int *arr ;
scanf("%d" , &n);
arr = (int*)malloc( n * sizeof(int));
for(ind = 0 ; ind < n ; ind++)
    scanf("%d" ,&arr[ind]);

int safe , count = 0;
safe = n ;
while(n!=0)
{
for(parent = n / 2 - 1;parent >=0 ; parent--)
{
    if(count % 2 == 0)
       maxHeapify(arr , &n , parent);
    else
       minHeapify(arr , &n , parent);
}
swap(arr , 0, --n);

count++;
```

```
}
for(ind = safe-1 ; ind >= 0 ; ind--)
     printf("%d " , arr[ind]);
return 0;
}
void swap(int*arr , int pos1,int pos2)
{
  int temp;
  temp = arr[pos1];
  arr[pos1]= arr[pos2];
  arr[pos2]= temp;

}
void maxHeapify(int*arr , int *size , int parent)
{
  int left,right;
  left= parent * 2+1;
  right = left + 1;
  // no child
    if(left >= *size)  return;
  // 1child
    if(right >= *size)
    {
       if(arr[parent] < arr[left])
       {
        swap(arr , left,parent);
        maxHeapify(arr , size , left);
       }
       return ;
    }
  // 2 child
  if(arr[parent] < arr[left] || arr[parent] < arr[right])
  {
      if(arr[left] > arr[right])
      {
    swap(arr, left,parent);
    maxHeapify(arr , size , left);
      }
      else
      {
    swap(arr , right,parent);
    maxHeapify(arr , size , right);
      }
  }
}
void minHeapify(int*arr , int *size , int parent)
{
  int left,right;
  left= parent * 2+1;
  right = left + 1;
  // no child
    if(left >= *size)  return;
  // 1child
    if(right >= *size)
    {
       if(arr[parent] > arr[left])
       {
        swap(arr , left,parent);
        maxHeapify(arr , size , left);
       }
       return ;
    }
  // 2 child
  if(arr[parent] > arr[left] || arr[parent] > arr[right])
  {
      if(arr[left] < arr[right])
      {
    swap(arr, left,parent);
    minHeapify(arr , size , left);
      }
      else
      {
    swap(arr , right,parent);
    minHeapify(arr , size , right);
      }
  }
}
```

```
}
for(ind = safe-1 ; ind >= 0 ; ind--)
     printf("%d " , arr[ind]);
return 0;
}
void swap(int*arr , int pos1,int pos2)
{
  int temp;
  temp = arr[pos1];
  arr[pos1]= arr[pos2];
  arr[pos2]= temp;

}
void maxHeapify(int*arr , int *size , int parent)
{
  int left,right;
  left= parent * 2+1;
  right = left + 1;
  // no child
    if(left >= *size)  return;
  // 1child
    if(right >= *size)
    {
       if(arr[parent] < arr[left])
       {
        swap(arr , left,parent);
        maxHeapify(arr , size , left);
       }
       return ;
    }
  // 2 child
  if(arr[parent] < arr[left] || arr[parent] < arr[right])
  {
      if(arr[left] > arr[right])
      {
    swap(arr, left,parent);
    maxHeapify(arr , size , left);
      }
      else
      {
    swap(arr , right,parent);
    maxHeapify(arr , size , right);
      }
  }
}
void minHeapify(int*arr , int *size , int parent)
{
  int left,right;
  left= parent * 2+1;
  right = left + 1;
  // no child
    if(left >= *size)  return;
  // 1child
    if(right >= *size)
    {
       if(arr[parent] > arr[left])
       {
        swap(arr , left,parent);
        maxHeapify(arr , size , left);
       }
       return ;
    }
  // 2 child
  if(arr[parent] > arr[left] || arr[parent] > arr[right])
  {
      if(arr[left] < arr[right])
      {
    swap(arr, left,parent);
    minHeapify(arr , size , left);
      }
      else
      {
    swap(arr , right,parent);
    minHeapify(arr , size , right);
      }
  }
}
```

Q5

Test Case

Input

```
434
```

Output

```
111 111 111 101
4
```

Weightage - 1

Input

```
10199
```

Output

```
10100 11 11 11 11 11 11 11 11 11
10
```

Weightage - 15

Input

```
42442
```

Output

```
11111 11111 11111 1111 1111 1111 1111 1111 1111 1111 1111 111
13
```

Weightage - 30

Input

```
4674
```

Output

```
1111 1111 1111 1111 111 111 1 1 1 1 1 1 1 1
14
```

Weightage - 10

Input

```
3654
```

Output

```
1111 1111 1111 111 111 11 11 11 11 11 11 11 11 11
14
```

Weightage - 20

Input

```
46754
```

Output

```
11111 11111 11111 11111 1111 1111 11 11 11 11 11 11 11 11
14
```

Weightage - 5

Input

```
16939
```

Output

```
11111 1111 1111 1111 1111 1111 111 111 11 11 11 11 1 1 1 1 1
19
```

Weightage - 1

Input

```
9990
```

Output

```
1111 1111 1111 1111 1111 1111 1111 1111 1101 1
10
```

Weightage - 18

Sample Input

```
32
```

Sample Output

```
11 11 10
3
```

Solution

```c
#include<stdio.h>
int nod(int num)
{
    int spare = 0 ,digit , power , newnum = 0 ;
    power = 1;
    while(num/power)
    {
       digit= (num / power)%10;
       spare = spare * 10 + 1;
       if(digit == 0 || digit == 9)
          newnum = 0 * power + newnum;
       else
          newnum = 1 * power + newnum;
       power *= 10;
    }

return (newnum == 0||spare<=num) ? spare : newnum;
}
int main()
{
int num , val , ctr , count = 0 ;;
scanf("%d" , &num);
do
{
   val = nod(num);
   if( val > num && num >= 10)
   {
       printf("%d" , num );
       count++;
       break;
   }
   if( num>=0 && num <= 9)
   {
       for(ctr = 1 ; ctr <= num ; ctr++)
       {
        printf("%d " , 1);
        count++;
       }
     break;
   }
   while( (num-val)>=0 && num)
   {
       num -= val;
       printf("%d " , val);
       count++;
   }

}while( num != 0 );
printf("\n%d" , count);
return 0;
}
```

```c
#include<stdio.h>
int nod(int num)
{
    int spare = 0 ,digit , power , newnum = 0 ;
    power = 1;
    while(num/power)
    {
       digit= (num / power)%10;
       spare = spare * 10 + 1;
       if(digit == 0 || digit == 9)
          newnum = 0 * power + newnum;
       else
          newnum = 1 * power + newnum;
       power *= 10;
    }

return (newnum == 0||spare<=num) ? spare : newnum;
}
int main()
{
int num , val , ctr , count = 0 ;;
scanf("%d" , &num);
do
{
   val = nod(num);
   if( val > num && num >= 10)
   {
       printf("%d" , num );
       count++;
       break;
   }
   if( num>=0 && num <= 9)
   {
       for(ctr = 1 ; ctr <= num ; ctr++)
       {
        printf("%d " , 1);
        count++;
       }
     break;
   }
   while( (num-val)>=0 && num)
   {
       num -= val;
       printf("%d " , val);
       count++;
   }

}while( num != 0 );
printf("\n%d" , count);
return 0;
}
```

Q6

Test Case

Input

```
6
```

Output

```
                  1   2   3   4   5   6
              12  11  10   9   8   7
          13  14  15  16  17  18
      24  23  22  21  20  19
```

Weightage - 20

Input

```
3
```

Output

```
        1   2   3
    6   5   4
7   8   9
```

Weightage - 20

Input

```
7
```

Output

```
                      1   2   3   4   5   6   7
                  14  13  12  11  10   9   8
              15  16  17  18  19  20  21
          28  27  26  25  24  23  22
```

Weightage - 20

Input

```
4
```

Output

```
            1   2   3   4
        8   7   6   5
    9  10  11  12
16  15  14  13
```

Weightage - 20

Input

```
9
```

Output

```
                            1   2   3   4   5   6   7   8   9
                        18  17  16  15  14  13  12  11  10
                    19  20  21  22  23  24  25  26  27
                36  35  34  33  32  31  30  29  28
```

Weightage - 20

Sample Input

```
5
```

Sample Output

```
                1   2   3   4   5
            10   9   8   7   6
        11  12  13  14  15
    20  19  18  17  16
```

Solution

```
#include<stdio.h>
int main()
{
    int row , col , N , count , val, spacecount;
    scanf("%d" , &N);
    for(row = 1, count = 0 , spacecount = (N-1); row <= N ; printf("\n") , row++,spacecount--)
    {
      for(col = 1 ; col <= spacecount ;printf("   "),col++);

      if(row % 2 == 0)
        {
           val = count * N ;
           for(col = 1 ; col <= N ; printf("%*d " , 2, val--),col++);
        }
      else
        {
           val = count * N + 1 ;
           count+=2;
           for(col = 1 ; col <= N ; printf("%*d " , 2,val++),col++);
        }

    }

    return 0;
}
```

```
#include<stdio.h>
int main()
{
    int row , col , N , count , val, spacecount;
    scanf("%d" , &N);
    for(row = 1, count = 0 , spacecount = (N-1); row <= N ; printf("\n") , row++,spacecount--)
    {
      for(col = 1 ; col <= spacecount ;printf("   "),col++);

      if(row % 2 == 0)
        {
           val = count * N ;
           for(col = 1 ; col <= N ; printf("%*d " , 2, val--),col++);
        }
      else
        {
           val = count * N + 1 ;
           count+=2;
           for(col = 1 ; col <= N ; printf("%*d " , 2,val++),col++);
        }

    }
```

```
    return 0;
}
```

Q7

Test Case

Input

```
8
hai hello how are you i am good
haihellohowareyouiamgood
```

Output

```
hai hello how are you i am good
```

Weightage - 15

Input

```
5
i love sum sung sumsung
ilovesumsung
```

Output

```
i love sumsung
```

Weightage - 5

Input

```
5
love india delhi in i
iloveindiaindelhi
```

Output

```
i love india in delhi
```

Weightage - 15

Input

```
7
i mobile phone love apple android i7
iloveapplei7phone
```

Output

```
i love apple i7 phone
```

Weightage - 15

Input

```
6
lovely love loves loved being bee
lovelyloveloveslovedbeing
```

Output

```
lovely love loves loved being
```

Weightage - 25

Input

```
5
wonder wonderful rain bow rainbow
wonderfulrainbow
```

Output

```
wonderful rainbow
```

Weightage - 10

Input

```
3
hai haai haaai
haaaihaai
```

Output

```
haaai haai
```

Weightage - 10

Input

```
3
lovely dear loving
lovelydear
```

Output

```
lovely dear
```

Weightage - 5

Sample Input

Sample Output

5 i like ice cream icecream ilikeicecream	i like icecream

Solution

```
#include<stdio.h>
int StrLen(char *str)
{
    int ind;
    for(ind = 0 ; str[ind] ; ind++);
return ind;
}
int strCmp(char *s1 , char *s2)
{
    int ind;
    for(ind = 0 ; s1[ind] && s2[ind] && s1[ind] == s2[ind] ; ind++);
return s1[ind] - s2[ind];
}
int main()
{
int N , ind , ind1 , len , safeind1, prevlen;
char safe , *start ;
scanf("%d" , &N);
char str[N][50];
char input[50];
for(ind = 0 ; ind < N ; ind++)
    scanf("%s" , str[ind]);
scanf("%s" , input);
start = input;
prevlen = -1;
for( ind =0 ; input[ind] ; ind++ , prevlen = - 1)
{
   for(ind1 = 0 ; ind1 < N ; ind1++)
   {
       len = StrLen(str[ind1]);
       safe = input[ind+len];
       input[ind+len] = 0;
       if( strCmp(input+ind , str[ind1]) == 0 && prevlen < len)
       {
           safeind1  = ind1;
           prevlen = len;
       }
       input[ind+len] = safe;
   }
   printf("%s " , str[safeind1]);
   ind = ind + StrLen(str[safeind1]) - 1;

}
return 0 ;
}
```

```
#include<stdio.h>
int StrLen(char *str)
{
    int ind;
    for(ind = 0 ; str[ind] ; ind++);
return ind;
}
int strCmp(char *s1 , char *s2)
{
    int ind;
    for(ind = 0 ; s1[ind] && s2[ind] && s1[ind] == s2[ind] ; ind++);
return s1[ind] - s2[ind];
}
int main()
{
int N , ind , ind1 , len , safeind1, prevlen;
char safe , *start ;
scanf("%d" , &N);
char str[N][50];
char input[50];
for(ind = 0 ; ind < N ; ind++)
    scanf("%s" , str[ind]);
scanf("%s" , input);
start = input;
prevlen = -1;
for( ind =0 ; input[ind] ; ind++ , prevlen = - 1)
```

```
{
   for(ind1 = 0 ; ind1 < N ; ind1++)
   {
       len = StrLen(str[ind1]);
       safe = input[ind+len];
       input[ind+len] = 0;
       if( strCmp(input+ind , str[ind1]) == 0 && prevlen < len)
       {
           safeind1  = ind1;
           prevlen = len;
       }
       input[ind+len] = safe;
   }
   printf("%s " , str[safeind1]);
   ind = ind + StrLen(str[safeind1]) - 1;

}
return 0 ;
}
```

Q8

Test Case

Input

```
4
```

Output

```
A B C D

D A B C
```

Weightage - 15

Input

```
4
```

Output

```
A B C D

D A B C
```

Weightage - 5

Input

```
5
```

Output

```
A B C D E

E A B C D
```

Weightage - 15

Input

```
2
```

Output

```
A B

B A
```

Weightage - 5

Input

```
7
```

Output

```
A B C D E F G

G A B C D E F
```

Weightage - 20

Input

```
9
```

Output

```
A B C D E F G H I

I A B C D E F G H
```

Weightage - 15

Input

```
11
```

Output

```
A B C D E F G H I J K
```

```
K A B C D E F G H I J
```

Weightage - 15

Input

```
6
```

Output

```
A B C D E F

F A B C D E
```

Weightage - 10

Sample Input

```
3
```

Sample Output

```
A B C

C A B
```

Solution

```
#include<stdio.h>
#include<malloc.h>
void setString(char *str , int N)
{
int ctr;
    for(ctr = 0 ; ctr < N ; ctr++)
        str[ctr] = ctr +65;
str[ctr] = 0;
}
void strRev(char *str)
{
    int start , end;
    char temp;
    for(end = 0 ; str[end] ; end++);
    for(start = 0 , --end ; start < end ; start++ , end--)
    {
        temp = str[start];
        str[start] = str[end];
        str[end] = temp;
    }
}
void strRotate(char *str)
{
    strRev(str);
    strRev(str + 1);
}
void strCopy(char *s1 , char *s2)
{
int ind;
    for(ind = 0 ; s2[ind] ; s1[ind] = s2[ind] , ind++);
  s1[ind] = 0 ;
}
int main()
{
int N , row , col;

scanf("%d" ,&N);
char arr[N][N+1];
char str[N+1];
setString(str , N);

  for(row = 0 ; row < N ; row++ , printf("\n"))
  {
     strCopy(arr[row] , str) ;
    for(col = 0 ; col < N ; col++)
       printf("%c " , arr[row][col]);
    printf("\n");
     strRotate(str);
  }


return 0;
}
```

```
#include<stdio.h>
#include<malloc.h>
void setString(char *str , int N)
{
int ctr;
    for(ctr = 0 ; ctr < N ; ctr++)
        str[ctr] = ctr +65;
str[ctr] = 0;
}
void strRev(char *str)
{
    int start , end;
    char temp;
    for(end = 0 ; str[end] ; end++);
    for(start = 0 , --end ; start < end ; start++ , end--)
    {
        temp = str[start];
        str[start] = str[end];
        str[end] = temp;
    }
}
void strRotate(char *str)
{
    strRev(str);
    strRev(str + 1);
}
void strCopy(char *s1 , char *s2)
{
int ind;
    for(ind = 0 ; s2[ind] ; s1[ind] = s2[ind] , ind++);
  s1[ind] = 0 ;
}
int main()
{
int N , row , col;

scanf("%d" ,&N);
char arr[N][N+1];
char str[N+1];
setString(str , N);

  for(row = 0 ; row < N ; row++ , printf("\n"))
  {
     strCopy(arr[row] , str) ;
    for(col = 0 ; col < N ; col++)
       printf("%c " , arr[row][col]);
    printf("\n");
     strRotate(str);
  }


return 0;
}
```

Q9

Test Case

Input

8

Output

1
1 1
2 1
1 2 1 1

Weightage - 5

Input

6

Output

1
1 1
2 1
1 2 1 1

Weightage - 10

Input

4

Output

1
1 1
2 1
1 2 1 1

Weightage - 5

Input

7

Output

1
1 1
2 1
1 2 1 1

Weightage - 10

Input

8

Output

1
1 1
2 1
1 2 1 1

Weightage - 10

Input

9

Output

1
1 1
2 1
1 2 1 1

Weightage - 20

Input

10

Output

1
1 1
2 1
1 2 1 1

Weightage - 20

Input

11

Output

1
1 1
2 1
1 2 1 1

Weightage - 20

Sample Input

5

Sample Output

1
1 1
2 1
1 2 1 1

Solution

```c
#include<stdio.h>
#include<malloc.h>
int main()
{

int N , row , col , ctr ,count , val , ind;
scanf("%d" ,&N);
int **arr;
arr = (int**) calloc(N , sizeof(int*));
for(ctr = 0 ; ctr < N ; ctr++)
    arr[ctr] = (int*)calloc(N*N/4 , sizeof(int));

arr[0][0] = 1;
ind = 0;
val = 1;
count = 0;
for(row = 0 ; row < N-1 ; row++ , ind = 0  )
{
    for(col = 0 ;  arr[row][col] && col < (N*N/4)-1 ; col++)
    {
        if(arr[row][col] == arr[row][col+1])
            count++;
        else if(arr[row][col+1] == 0 && count == 0) // last mismatch
        {
           arr[row+1][ind] = 1;
           arr[row+1][ind+1] = arr[row][col];
           count = 0;
           ind += 2;
        }
        else
        {
           arr[row+1][ind] = count + 1;
           arr[row+1][ind+1] = arr[row][col];
           count = 0;
           ind += 2;
        }
    }
}
for(row = 0 ; row < N ; row++ , printf("\n"))
{
    for(col = 0 ; arr[row][col] && col < N*N/4 ; col++)
         printf("%d " , arr[row][col]);
}
return 0;
}



#include<stdio.h>
#include<malloc.h>
int main()
{

int N , row , col , ctr ,count , val , ind;
scanf("%d" ,&N);
int **arr;
arr = (int**) calloc(N , sizeof(int*));
for(ctr = 0 ; ctr < N ; ctr++)
    arr[ctr] = (int*)calloc(N*N/4 , sizeof(int));

arr[0][0] = 1;
ind = 0;
val = 1;
count = 0;
for(row = 0 ; row < N-1 ; row++ , ind = 0  )
{
    for(col = 0 ;  arr[row][col] && col < (N*N/4)-1 ; col++)
    {
        if(arr[row][col] == arr[row][col+1])
            count++;
        else if(arr[row][col+1] == 0 && count == 0) // last mismatch
        {
           arr[row+1][ind] = 1;
           arr[row+1][ind+1] = arr[row][col];
           count = 0;
           ind += 2;
        }
        else
        {
           arr[row+1][ind] = count + 1;
```

```
            arr[row+1][ind+1] = arr[row][col];
            count = 0;
            ind += 2;
          }
      }
  }
  for(row = 0 ; row < N ; row++ , printf("\n"))
  {
      for(col = 0 ; arr[row][col] && col < N*N/4 ; col++)
           printf("%d " , arr[row][col]);
  }
  return 0;
  }
```

Q10

Test Case

Input

```
a12b3d4
```

Output

```
aaaaaaaaaaaabbbdddd
```

Weightage - 10

Input

```
a100b3c12
```

Output

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

Weightage - 10

Input

```
a76b23c10
```

Output

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

Weightage - 20

Input

```
x7y3d8
```

Output

```
xxxxxxxyyyddddddd
```

Weightage - 15

Input

```
l123a3
```

Output

```
lllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllll
```

Weightage - 20

Input

```
t6h7j9k3h6
```

Output

```
ttttttthhhhhhhjjjjjjjjjkkkhhhhhh
```

Weightage - 10

Input

```
j7b7m8v7b8
```

Output

```
jjjjjjjbbbbbbbmmmmmmmmvvvvvvvbbbbbbbb
```

Weightage - 5

Input

Output

```
j8b8n9j7m7v87
```

```
jjjjjjjjbbbbbbbbnnnnnnnnnjjjjjjjmmmmmmmvvvvvvvvvvvvvvvvvvvvvvvvvvv
```

Weightage - 10

Sample Input

Sample Output

```
a2b4c6
```

```
aabbbbccccccc
```

Solution

```
#include<stdio.h>
int main()
{
char str[200] , ch;
int ind , count;
scanf("%s" , str);
int len , start;
for(len = 0 ; str[len] ; len++);
start = len ;
ind = count =  0;
while( ind < len )
{
    ch = str[ind++];
    while(str[ind] >= '0' && str[ind] <= '9' )
        count = count * 10 + (str[ind++] - '0');
    while(count)
    {
        str[start++] = ch;
        count--;
    }
}
for(ind = len ; ind < start ; ind++)
    str[ind - len] =str[ind];
str[ind - len] = 0;
printf("%s" , str);


return 0;
}
```

```
#include<stdio.h>
int main()
{
char str[200] , ch;
int ind , count;
scanf("%s" , str);
int len , start;
for(len = 0 ; str[len] ; len++);
start = len ;
ind = count =  0;
while( ind < len )
{
    ch = str[ind++];
    while(str[ind] >= '0' && str[ind] <= '9' )
        count = count * 10 + (str[ind++] - '0');
    while(count)
    {
        str[start++] = ch;
        count--;
    }
}
for(ind = len ; ind < start ; ind++)
    str[ind - len] =str[ind];
str[ind - len] = 0;
printf("%s" , str);


return 0;
}
```

Q11

Test Case

Input

Output

```
203
```

```
two hundred and three
```

Weightage - 20

Input

Output

```
0
```

```
zero
```

Weightage - 5

Input

Output

```
545
```

```
five hundred and forty five
```

Weightage - 15

Input

Output

999	nine hundred and ninety nine

Weightage - 15

Input	Output
12	twelve

Weightage - 5

Input	Output
34	thirty four

Weightage - 5

Input	Output
550	five hundred and fifty

Weightage - 15

Input	Output
1	one

Weightage - 20

Sample Input	Sample Output
234	two hundred and thirty four

Sample Input	Sample Output
200	two hundred only

Solution

```
#include<stdio.h>
int main()
{
char one[21][10]={"zero" , "one" , "two" , "three" , "four" , "five" , "six" , "seven" ,
 "eight" , "nine" ,"" , "eleven" , "twelve" , "thirteen" , "fourteen" , "fifteen" , "sixteen" ,
 "seventeen" , "eighteen" , "nineteen"};
char tens[10][9]={"" , "ten","twenty","thirty","forty","fifty","sixty","seventy",
"eighty","ninety"};
int num;
scanf("%d" , &num);
if(num >=0 && num <= 99)
{
    if( num >= 0 && num <=19) printf("%s" , one[num]);
    else
        {
           printf("%s" , tens [ num / 10 ]);
           if(num % 10 ) printf(" %s" , one[ num % 10]);
        }
}
else if( num / 100)
{
 printf("%s hundred" , one[ num / 100 ] );
```

```
 if( num % 100 == 0) printf(" only");
 else
  {
      int rem;
      rem = num % 100;
      if( rem >= 1 && rem <=19)
         printf(" and %s" , one[rem]);
      else
        {
           printf(" and %s" , tens [ rem / 10 ]);
           if(num % 10 ) printf(" %s" , one[ rem % 10]);
        }
  }
}
return 0;
}
```

```
#include<stdio.h>
int main()
{
char one[21][10]={"zero" , "one" , "two" , "three" , "four" , "five" , "six" , "seven" ,
 "eight" , "nine" ,"" , "eleven" , "twelve" , "thirteen" , "fourteen" , "fifteen" , "sixteen" ,
 "seventeen" , "eighteen" , "nineteen"};
char tens[10][9]={"" , "ten","twenty","thirty","forty","fifty","sixty","seventy",
"eighty","ninety"};
int num;
scanf("%d" , &num);
if(num >=0 && num <= 99)
{
    if( num >= 0 && num <=19) printf("%s" , one[num]);
    else
        {
           printf("%s" , tens [ num / 10 ]);
           if(num % 10 ) printf(" %s" , one[ num % 10]);
        }
}
else if( num / 100)
{
 printf("%s hundred" , one[ num / 100 ] );
 if( num % 100 == 0) printf(" only");
 else
  {
      int rem;
      rem = num % 100;
      if( rem >= 1 && rem <=19)
         printf(" and %s" , one[rem]);
      else
        {
           printf(" and %s" , tens [ rem / 10 ]);
           if(num % 10 ) printf(" %s" , one[ rem % 10]);
        }
  }
}
return 0;
}
```

Q12

Test Case

Input

```
3
```

Output

```
3 3 3 3 3
3 2 2 2 3
3 2 1 2 3
3 2 2 2 3
```

Weightage - 10

Input

```
7
```

Output

```
7 7 7 7 7 7 7 7 7 7 7 7 7
7 6 6 6 6 6 6 6 6 6 6 6 7
7 6 5 5 5 5 5 5 5 5 6 7
7 6 5 4 4 4 4 4 4 5 6 7
```

Weightage - 20

Input

Output

2	2 2 2 2 1 2 2 2 2

Weightage - 10

Input **Output**

4	4 4 4 4 4 4 4 4 3 3 3 3 3 4 4 3 2 2 2 3 4 4 3 2 1 2 3 4

Weightage - 10

Input **Output**

9	9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 9 9 8 7 7 7 7 7 7 7 7 7 7 7 7 7 8 9 9 8 7 6 6 6 6 6 6 6 6 6 6 6 7 8 9

Weightage - 20

Input **Output**

8	8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 7 6 6 6 6 6 6 6 6 6 6 6 7 8 8 7 6 5 5 5 5 5 5 5 5 5 6 7 8

Weightage - 20

Input **Output**

6	6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 6 6 5 4 4 4 4 4 4 4 5 6 6 5 4 3 3 3 3 3 4 5 6

Weightage - 10

Sample Input **Sample Output**

5	5 5 5 5 5 5 5 5 5 5 4 4 4 4 4 4 4 5 5 4 3 3 3 3 3 4 5 5 4 3 2 2 2 3 4 5

Solution

```
#include<stdio.h>
int setNum_Pow(int n , int *pow )
{
    int num = 0 , ind ;
    *pow = 1;
    for(ind = 0 ; ind < n ; num = num * 10 + n , (*pow) = *pow *  10 + 1 , ind++);
    (*pow) /= 100;
return num/10;
}
int reverse(int num)
{
    int revnum = 0 ;
    while(num)
    {
        revnum = revnum * 10 + (num % 10);
        num /= 10;
    }
return revnum;
}
int main()
{
int N , num , row , count , col , pow , revnum, POW;
scanf("%d" , &N);
num = setNum_Pow(N-1 , &pow );
for(row= 1 ; row <= (2*N-1) ; row++)
    printf("%d " , N);
printf("\n");

  count = N-1;
```

```
  for(row = 2 ; row <= N ; row++ , printf("\n"))
  {
    printf("%d " ,N);
    POW = 1;
     revnum = reverse(num);

    while(revnum / POW)
    {
        printf("%d " , (revnum/POW) % 10);
        POW *= 10;
    }
    printf("%d " , count--);
    POW = 1;
    while(num / POW)
    {
        printf("%d " , (num/POW) % 10);
        POW *= 10;
    }
    printf("%d " ,N);
    pow /= 10;
    num = num - pow ;
 }
// second half
 count = 2 ;
  for(row = 2 ; row <= N ; row++ , printf("\n"))
  {
    printf("%d " ,N);
    POW = 1;
     revnum = reverse(num);

    while(revnum / POW)
    {
        printf("%d " , (revnum/POW) % 10);
        POW *= 10;
    }
    printf("%d " , count++);
    POW = 1;
    while(num / POW)
    {
        printf("%d " , (num/POW) % 10);
        POW *= 10;
    }
    printf("%d " ,N);
    pow = pow * 10 + 1;
    num = num + pow ;
 }
return 0;
}



#include<stdio.h>
int setNum_Pow(int n , int *pow )
{
    int num = 0 , ind ;
    *pow = 1;
    for(ind = 0 ; ind < n ; num = num * 10 + n , (*pow) = *pow *  10 + 1 , ind++);
    (*pow) /= 100;
return num/10;
}
int reverse(int num)
{
    int revnum = 0 ;
    while(num)
    {
        revnum = revnum * 10 + (num % 10);
        num /= 10;
    }
return revnum;
}
int main()
{
int N , num , row , count , col , pow , revnum, POW;
scanf("%d" , &N);
num = setNum_Pow(N-1 , &pow );
for(row= 1 ; row <= (2*N-1) ; row++)
    printf("%d " , N);
printf("\n");


  count = N-1;
  for(row = 2 ; row <= N ; row++ , printf("\n"))
```

```
  {
    printf("%d " ,N);
    POW = 1;
     revnum = reverse(num);

    while(revnum / POW)
    {
        printf("%d " , (revnum/POW) % 10);
        POW *= 10;
    }
    printf("%d " , count--);
    POW = 1;
    while(num / POW)
    {
        printf("%d " , (num/POW) % 10);
        POW *= 10;
    }
    printf("%d " ,N);
    pow /= 10;
    num = num - pow ;
 }
// second half
 count = 2 ;
  for(row = 2 ; row <= N ; row++ , printf("\n"))
  {
    printf("%d " ,N);
    POW = 1;
     revnum = reverse(num);

    while(revnum / POW)
    {
        printf("%d " , (revnum/POW) % 10);
        POW *= 10;
    }
    printf("%d " , count++);
    POW = 1;
    while(num / POW)
    {
        printf("%d " , (num/POW) % 10);
        POW *= 10;
    }
    printf("%d " ,N);
    pow = pow * 10 + 1;
    num = num + pow ;
 }
return 0;
}
```

Q13

Test Case

Input

```
12
47 85 2 8 92 35 92 89 20 58 587 24
```

Output

```
587 8 92 35 92 58 47 85 20 89 2 24
```

Weightage - 15

Input

```
7
1 2 3 4 5 6 7
```

Output

```
7 2 5 4 3 6 1
```

Weightage - 10

Input

```
7
10 20 30 40 50 60 70
```

Output

```
70 20 50 40 30 60 10
```

Weightage - 10

Input

Output

8 487 7824 623 75 85 42 24 6745	623 42 487 75 85 7824 24 6745

Weightage - 20

Input	Output
4 4643 63 7842 745	7842 63 4643 745

Weightage - 10

Input	Output
5 4387 75348 2364 7856 23	4387 7856 2364 75348 23

Weightage - 10

Input	Output
8 347 743 632 8 25 854 452 8	632 8 452 743 347 854 25 8

Weightage - 15

Input	Output
6 375 784 834 98246 7542 87	7542 784 834 98246 375 87

Weightage - 10

Sample Input	Sample Output
9 23 7 8 30 18 12 6 28 16	23 7 18 12 16 28 8 30 6

Solution

```
#include<stdio.h>
void ASC_BubbleSort(int*arr , int N , int start)
{
    int ind , flag , temp;
do
{
    for(ind = start , flag = 0; ind < N- 2 ; ind+=2)
    {
        if(arr[ind] > arr[ind+2])
        {
            flag = 1;
            temp = arr[ind];
            arr[ind] = arr[ind+2];
            arr[ind+2] = temp;
        }
    }
}while(flag == 1);

}
void DEC_BubbleSort(int*arr , int N , int start)
{
    int ind , flag , temp;
do
{
    for(ind = start , flag = 0; ind < N- 2 ; ind+=2)
    {
        if(arr[ind] < arr[ind+2])
        {
            flag = 1;
```

```
#include<stdio.h>
void ASC_BubbleSort(int*arr , int N , int start)
{
    int ind , flag , temp;
do
{
    for(ind = start , flag = 0; ind < N- 2 ; ind+=2)
    {
        if(arr[ind] > arr[ind+2])
        {
            flag = 1;
            temp = arr[ind];
            arr[ind] = arr[ind+2];
            arr[ind+2] = temp;
        }
    }
}while(flag == 1);

}
void DEC_BubbleSort(int*arr , int N , int start)
{
    int ind , flag , temp;
do
{
    for(ind = start , flag = 0; ind < N- 2 ; ind+=2)
    {
        if(arr[ind] < arr[ind+2])
        {
            flag = 1;
```

```
            temp = arr[ind];
            arr[ind] = arr[ind+2];
            arr[ind+2] = temp;
        }
    }
}while(flag == 1);

}
int main()
{
    int N, ind;
    scanf("%d" ,&N);
    int arr[N];
    for(ind = 0 ; ind < N ; scanf("%d" , &arr[ind++]));
    ASC_BubbleSort(arr , N-1 , 1);
    DEC_BubbleSort(arr , N , 0);
    for(ind = 0 ; ind < N ; printf("%d " , arr[ind++]));
}
```

```
            temp = arr[ind];
            arr[ind] = arr[ind+2];
            arr[ind+2] = temp;
        }
    }
}while(flag == 1);

}
int main()
{
    int N, ind;
    scanf("%d" ,&N);
    int arr[N];
    for(ind = 0 ; ind < N ; scanf("%d" , &arr[ind++]));
    ASC_BubbleSort(arr , N-1 , 1);
    DEC_BubbleSort(arr , N , 0);
    for(ind = 0 ; ind < N ; printf("%d " , arr[ind++]));
}
```

Q14

Test Case

Input

```
hai
```

Output

```
h    i
  a
h    i
```

Weightage - 10

Input

```
printthepattern
```

Output

```
p                                                   n
    r                                           r
        i                                   e
                n                       t
```

Weightage - 20

Input

```
hello
```

Output

```
h           o
    e    l
       l
    o    l
```

Weightage - 10

Input

```
wonderful
```

Output

```
w                           l
    o                   u
        n           f
            d   n
```

Weightage - 10

Input

```
cleartheworld
```

Output

```
c                                       d
    l                               l
        e                       r
            a               e
```

Weightage - 20

Input

```
forgive
```

Output

```
f                   e
    o           v
        r   i
            g
```

Weightage - 10

Input

```
examlyy
```

Output

```
e                   y
    x           y
```

```
        a    l
```

Weightage - 10

Input

```
zohoround
```

Output

```
z                         d
    o                   n
        h           u
```

Weightage - 10

Sample Input

```
welcome
```

Sample Output

```
w                   e
    e           m
        l   o
```

Solution

```
#include<stdio.h>
int main()
{
    char str[100];
scanf("%s" , str);
int len , space1 , space2 , row , ctr,start,end;
for(len = 0 ; str[len] ; len++);
space2 = len - 2;
space1 = 0;
start = 0;
end = len -  1;
for(row = 1 ; row <= (len/2+1) && start < end; row++, printf("\n"), space1++ , space2-=2)
{
    for(ctr = 1 ; ctr <= space1 ; printf("   "), ctr++);
    printf("%c",str[start++]);
    for(ctr = 1 ; ctr <= space2 ; printf("   "), ctr++);
    printf("%c", str[end--]);
}
    for(ctr = 1 ; ctr <space1  ; printf("   "), ctr++);
    printf("  %c",str[start++]);
space2 = 1;
space1 = (len/2)-1;
start--;
printf("\n");
for(row = 1 ; row < (len/2+1); row++, printf("\n"), space1-- , space2+=2)
{
    for(ctr = 1 ; ctr <= space1 ; printf("   "), ctr++);
    printf("%c",str[--start]);
    for(ctr = 1 ; ctr <= space2 ; printf("   "), ctr++);
    printf("%c", str[++end]);
}
return 0;
}
```

```
#include<stdio.h>
int main()
{
    char str[100];
    scanf("%s" , str);
    int len , space1 , space2 , row , ctr,start,end;
    for(len = 0 ; str[len] ; len++);
    space2 = len - 2;
    space1 = 0;
    start = 0;
    end = len -  1;
    for(row = 1 ; row <= (len/2+1) && start < end; row++, printf("\n"), space1++ , space2-=2)
    {
        for(ctr = 1 ; ctr <= space1 ; printf("   "), ctr++);
        printf("%c",str[start++]);
        for(ctr = 1 ; ctr <= space2 ; printf("   "), ctr++);
        printf("%c", str[end--]);
    }
        for(ctr = 1 ; ctr <space1  ; printf("   "), ctr++);
        printf("  %c",str[start++]);
    space2 = 1;
    space1 = (len/2)-1;
    start--;
```

```
    printf("\n");
    for(row = 1 ; row < (len/2+1); row++, printf("\n"), space1-- , space2+=2)
    {
        for(ctr = 1 ; ctr <= space1 ; printf("   "), ctr++);
        printf("%c",str[--start]);
        for(ctr = 1 ; ctr <= space2 ; printf("   "), ctr++);
        printf("%c", str[++end]);
    }
    return 0;
}
```

Q15

Test Case

Input

```
haihellohaihello
ello
```

Output

```
4
```

Weightage - 20

Input

```
wonder
ond
```

Output

```
1
```

Weightage - 5

Input

```
haihello
how
```

Output

```
-1
```

Weightage - 10

Input

```
dsjkfsjdkvbsdkbv
sjdkvb
```

Output

```
5
```

Weightage - 10

Input

```
jdfkbgvkjdfbvkjdfbvkjdfbjdfkbvkjdfb
vkjdfbvkjdfb
```

Output

```
6
```

Weightage - 15

Input

```
dbfvdskjbvdskbvksdbvsdvbdshbvdksv
vbdshbvdksv
```

Output

```
22
```

Weightage - 15

Input

```
dskbfvkjdgbvkfdASVCAHKSVHCsjgvdwiudytiWADC
ASVCAHKSVHC
```

Output

```
15
```

Weightage - 15

Input

Output

DKJVBDSKBVKDSVBKJDS KBVKDSVBKJ	7

Weightage - 10

Sample Input

Sample Output

thistest123string 123	8

Solution

```
#include<stdio.h>
int main()
{
   char str[200];
   char substr[200];
   scanf("%s %s", str , substr);
   int ind , ind1 , len , sublen;
   ind = 0 ;
   for(len = 0 ; str[len] ; len++);
   for(sublen = 0 ; substr[sublen] ; sublen++);
   if(sublen > len)
       printf("-1");
  else
  {
   while(str[ind])
   {
       if( str[ind] == substr[0])
       {
           ind++;
           ind1 = 1;

           while(str[ind] && substr[ind1] && str[ind] == substr[ind1])
           {
               ind1++;
               ind++;
           }
          if(substr[ind1] == 0 || str[ind] == 0)
            break;
       }
       else
        ind++;
   }
   if(substr[ind1] == 0)
     printf("%d", ind - sublen);
   else
    printf("-1");
  }
return 0;
}
```

```
#include<stdio.h>
int main()
{
   char str[200];
   char substr[200];
   scanf("%s %s", str , substr);
   int ind , ind1 , len , sublen;
   ind = 0 ;
   for(len = 0 ; str[len] ; len++);
   for(sublen = 0 ; substr[sublen] ; sublen++);
   if(sublen > len)
       printf("-1");
  else
  {
   while(str[ind])
   {
       if( str[ind] == substr[0])
       {
           ind++;
           ind1 = 1;

           while(str[ind] && substr[ind1] && str[ind] == substr[ind1])
           {
```

```
                ind1++;
                ind++;

            }
           if(substr[ind1] == 0 || str[ind] == 0)
              break;
        }
        else
         ind++;
    }
    if(substr[ind1] == 0)
      printf("%d", ind - sublen);
    else
     printf("-1");
  }
return 0;
}
```

Q16

Test Case

Input

```
7
4 7 8 10 12 30 35
9
1 3 4 5 7 8 11 13 22
```

Output

```
1 3 4 5 7 8 10 11 12 13 22 30 35
```

Weightage - 15

Input

```
10
1 2 3 4 5 6 7 8 9 10
10
1 2 3 11 12 13 14 15 16 17
```

Output

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
```

Weightage - 15

Input

```
5
1 2 3 4 5
5
1 2 3 4 5
```

Output

```
1 2 3 4 5
```

Weightage - 5

Input

```
8
20 22 34 56 78 90 120 200
5
35 80 99 125 250
```

Output

```
20 22 34 35 56 78 80 90 99 120 125 200 250
```

Weightage - 20

Input

```
7
22 33 44 55 66 77 88
10
11 20 22 30 33 40 44 50 55 60
```

Output

```
11 20 22 30 33 40 44 50 55 60 66 77 88
```

Weightage - 20

Input

```
10
1 2 3 4 5 6 7 8 9 10
15
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Output

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Weightage - 15

Input

```
7
578 689 789 900 1200 1345 1567

5
```

Output

```
578 689 789 900 1200 1345 1567 2000 2500
```

Weightage - 10

Sample Input

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

Sample Output

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

Solution

```
#include<stdio.h>
#include<malloc.h>
int main()
{

int N1 , N2, ind, ind1;
scanf("%d", &N1);
int arr[N1];
for(ind = 0 ; ind < N1 ;  scanf("%d" , &arr[ind++]));
scanf("%d" ,&N2);
int arr1[N2];
int *newarr , newind ;
newarr = (int*)calloc(N1+N2 , sizeof(int));
for(ind = 0 ; ind < N2 ;  scanf("%d" , &arr1[ind++]));
ind = ind1 = newind = 0 ;

while(ind < N1 && ind1 < N2)
{
   if(arr[ind] == arr1[ind1])
   {
       newarr[newind++] = arr[ind];
       ind++;
       ind1++;
   }
   else if( arr[ind] < arr1[ind1])
      newarr[newind++] = arr[ind++];
   else if(arr[ind] > arr1[ind1])
      newarr[newind++] = arr1[ind1++];
}
while(ind == N1 && ind1 < N2)
  newarr[newind++] = arr1[ind1++];
while(ind1 == N2 && ind < N1)
  newarr[newind++] = arr[ind++];
for(ind = 0 ; ind < newind ; ind++)
     printf("%d ", newarr[ind]);

return 0;
}
```

```
#include<stdio.h>
#include<malloc.h>
int main()
{

int N1 , N2, ind, ind1;
scanf("%d", &N1);
int arr[N1];
for(ind = 0 ; ind < N1 ;  scanf("%d" , &arr[ind++]));
scanf("%d" ,&N2);
int arr1[N2];
int *newarr , newind ;
newarr = (int*)calloc(N1+N2 , sizeof(int));
for(ind = 0 ; ind < N2 ;  scanf("%d" , &arr1[ind++]));
ind = ind1 = newind = 0 ;

while(ind < N1 && ind1 < N2)
{
   if(arr[ind] == arr1[ind1])
   {
       newarr[newind++] = arr[ind];
       ind++;
       ind1++;
   }
   else if( arr[ind] < arr1[ind1])
      newarr[newind++] = arr[ind++];
   else if(arr[ind] > arr1[ind1])
      newarr[newind++] = arr1[ind1++];
}
while(ind == N1 && ind1 < N2)
  newarr[newind++] = arr1[ind1++];
while(ind1 == N2 && ind < N1)
  newarr[newind++] = arr[ind++];
for(ind = 0 ; ind < newind ; ind++)
     printf("%d ", newarr[ind]);

return 0;
}
```

Q17

Test Case

Input

```
((ab)(c+d))
```

Output

```
((ab)(c+d))
```

Weightage - 15

Input

```
((ab)(cd)(((de)
```

Output

```
(ab)(cd)(de)
```

Weightage - 20

Input

```
((((((((((((((((ab)
```

Output

```
(ab)
```

Weightage - 10

Input

```
(ab)))))))))))
```

Output

```
(ab)
```

Weightage - 10

Input

```
(((((((((ab)((((((cd)(ef)
```

Output

```
(ab)(cd)(ef)
```

Weightage - 15

Input

```
(ab)))))(cd))))))(ef)
```

Output

```
(ab)(cd)(ef)
```

Weightage - 20

Input

```
(ab)))))))))(cd))
```

Output

```
(ab)(cd)
```

Weightage - 10

Sample Input

```
((abc)((de))
```

Sample Output

```
(abc)((de))
```

Sample Input

```
(((ab)
```

Sample Output

```
(ab)
```

Solution

```
#include<stdio.h>
#include<malloc.h>
int main()
{
    char str[50];
    scanf("%s" , str);
    int *arr , *close , ctop ,len , top , ind , safe , hash , nonhash;
    for(len = 0 ; str[len] ; len++);
    arr = (int*)calloc(len , sizeof(int));
    close = (int*)calloc(len , sizeof(int));
    ctop = top = -1;
    safe = 1;
    for(ind = 0 ; str[ind] ; ind++)
    {
        if(str[ind] == '(')
           arr[++top] = ind ;
        else if(str[ind] == ')')
        {
            if( top == -1)
            {
                close[++ctop] = ind;
                continue;
            }

            arr[top--] = -1;
        }

    }
    for(ind = 0 ; top >= 0 && ind <= top ; ind++)
```

```
        str[arr[ind]] = '#';
    for(ind = 0 ; ctop >= 0 && ind <= ctop ; ind++)
        str[close[ind]] = '#';
     hash = nonhash = 0;
     while(str[nonhash] && str[nonhash] == '#')
     {
         if(str[++nonhash] != '#')
             break;
     }
     while(str[nonhash])
     {
         while(str[nonhash] && str[nonhash] == '#')
              nonhash++;
         str[hash] = str[nonhash];
         hash++;
         nonhash++;
     }
     str[hash] = 0;
     printf("%s" , str);
return 0;
}
```

```
#include<stdio.h>
#include<malloc.h>
int main()
{
    char str[50];
    scanf("%s" , str);
    int *arr , *close , ctop ,len , top , ind , safe , hash , nonhash;
    for(len = 0 ; str[len] ; len++);
    arr = (int*)calloc(len , sizeof(int));
    close = (int*)calloc(len , sizeof(int));
    ctop = top = -1;
    safe = 1;
    for(ind = 0 ; str[ind] ; ind++)
    {
        if(str[ind] == '(')
           arr[++top] = ind ;
        else if(str[ind] == ')')
        {
            if( top == -1)
            {
                close[++ctop] = ind;
                continue;
            }

            arr[top--] = -1;
        }

    }
    for(ind = 0 ; top >= 0 && ind <= top ; ind++)
         str[arr[ind]] = '#';
    for(ind = 0 ; ctop >= 0 && ind <= ctop ; ind++)
        str[close[ind]] = '#';
     hash = nonhash = 0;
     while(str[nonhash] && str[nonhash] == '#')
     {
         if(str[++nonhash] != '#')
             break;
     }
     while(str[nonhash])
     {
         while(str[nonhash] && str[nonhash] == '#')
              nonhash++;
         str[hash] = str[nonhash];
         hash++;
         nonhash++;
     }
     str[hash] = 0;
     printf("%s" , str);
return 0;
}
```

Q18

Test Case

Input **Output**

```
7
20 35 57 30 56 87 30
10
```

```
33
```

Weightage - 10

Input

Output

```
12
67 743 73 634 89 734 9 76 90 36 65 34
12
```

```
226
```

Weightage - 20

Input

Output

```
10
10 20 30 40 50 60 70 80 90 100
10
```

```
55
```

Weightage - 10

Input

Output

```
5
4387 78 89 87965 57
100
```

```
927
```

Weightage - 20

Input

Output

```
7
438 879 56 123 8421 853 892
400
```

```
35
```

Weightage - 10

Input

Output

```
7
54378 8953 426 85 8964 2 9000
1000
```

```
85
```

Weightage - 15

Input

Output

```
10
489 853 843 835 895 89 24 8953 853 8935
853
```

```
31
```

Weightage - 15

Sample Input

Sample Output

```
6
5 8 10 13 6 2
3
```

```
17
```

Solution

```
#include<stdio.h>
int main()
{

int N , threshold , ind, count = 0;
scanf("%d" , &N);
int arr[N];
for(ind = 0 ; ind < N ; scanf("%d" , &arr[ind++]));
scanf("%d" , &threshold);

  for(ind = 0 ; ind < N ; ind++)
```

```
      count += (arr[ind]/ threshold) + ((arr[ind] % threshold)==0 ? 0 : 1);

printf("%d " ,count);
return 0;
}
```

```
#include<stdio.h>
int main()
{

int N , threshold , ind, count = 0;
scanf("%d" , &N);
int arr[N];
for(ind = 0 ; ind < N ; scanf("%d" , &arr[ind++]));
scanf("%d" , &threshold);

  for(ind = 0 ; ind < N ; ind++)
      count += (arr[ind]/ threshold) + ((arr[ind] % threshold)==0 ? 0 : 1);

printf("%d " ,count);
return 0;
}
```

Q19

Test Case

Input

```
7
```

Output

```
1 8 14 19 23 26 28
2 9 15 20 24 27
3 10 16 21 25
4 11 17 22
```

Weightage - 5

Input

```
8
```

Output

```
1 9 16 22 27 31 34 36
2 10 17 23 28 32 35
3 11 18 24 29 33
4 12 19 25 30
```

Weightage - 20

Input

```
10
```

Output

```
1 11 20 28 35 41 46 50 53 55
2 12 21 29 36 42 47 51 54
3 13 22 30 37 43 48 52
4 14 23 31 38 44 49
```

Weightage - 10

Input

```
4
```

Output

```
1 5 8 10
2 6 9
3 7
4
```

Weightage - 5

Input

```
6
```

Output

```
1 7 12 16 19 21
2 8 13 17 20
3 9 14 18
4 10 15
```

Weightage - 10

Input

```
2
```

Output

```
1 3
2
```

Weightage - 5

Input

```
9
```

Output

```
1  10  18  25  31  36  40  43  45
2  11  19  26  32  37  41  44
3  12  20  27  33  38  42
4  13  21  28  34  39
```

Weightage - 20

Input

```
14
```

Output

```
1  15  28  40  51  61  70  78  85  91  96  100  103  105
2  16  29  41  52  62  71  79  86  92  97  101  104
3  17  30  42  53  63  72  80  87  93  98  102
4  18  31  43  54  64  73  81  88  94  99
```

Weightage - 25

Sample Input

```
5
```

Sample Output

```
1  6  10  13  15
2  7  11  14
3  8  12
4  9
```

Solution

```
#include<stdio.h>
int main()
{
int N , row ,col , val , counter;
scanf("%d" , &N);
for(row = N ; row > 0 ; row-- , printf("\n"))
{
    printf("%d ", N-row+1);
    for(counter = N-1 ,val = 1 ,val = val + counter + (N - row+1 ), col = 1 ; col < row ; col++ , counter--)
    {

        printf("%d " , val);
        val += counter;
    }
}



return 0;
}
```

```
#include<stdio.h>
int main()
{
int N , row ,col , val , counter;
scanf("%d" , &N);
for(row = N ; row > 0 ; row-- , printf("\n"))
{
    printf("%d ", N-row+1);
    for(counter = N-1 ,val = 1 ,val = val + counter + (N - row+1 ), col = 1 ; col < row ; col++ , counter--)
    {

        printf("%d " , val);
        val += counter;
    }
}



return 0;
}
```

Q20

Test Case

Input

```
hjdhjjduk
hjdhjjdua
```

Output

```
k,a
```

Weightage - 10

Input

```
dhjcdjcjdhvcjhdvchjdvs
dhjedjcjdhvfjhdvhhjdvs
```

Output

```
c,e c,f c,h
```

Weightage - 20

Input

```
sakckjsgckjdscg
sakckjsgckjdsca
```

Output

```
g,a
```

Weightage - 10

Input

```
dsjbvdjskbvj
dsjbddjgkbgj
```

Output

```
v,d s,g v,g
```

Weightage - 10

Input

```
sdjkcbdskjcb
sdjKcbDskJcb
```

Output

```
k,K d,D j,J
```

Weightage - 15

Input

```
askjcbgdskjcb
ashjcjjjskjcb
```

Output

```
k,h b,j g,j d,j
```

Weightage - 20

Input

```
dkjvbkjbvbjkf
dkjvfkjhvkjkf
```

Output

```
b,f b,h b,k
```

Weightage - 15

Sample Input

```
abcdefgh
abdfhjfb
```

Sample Output

```
c,d d,f e,h f,j g,f h,b
```

Solution

```
#include<stdio.h>
int main()
{
char str1[50];
char str2[50];
int ind;
scanf("%s %s" , str1 , str2);

for(ind = 0 ; str1[ind] ; ind++)
{
    if(str1[ind] != str2[ind])
        printf("%c,%c ", str1[ind] , str2[ind]);
}
```

```
#include<stdio.h>
int main()
{
char str1[50];
char str2[50];
int ind;
scanf("%s %s" , str1 , str2);

for(ind = 0 ; str1[ind] ; ind++)
{
    if(str1[ind] != str2[ind])
        printf("%c,%c ", str1[ind] , str2[ind]);
}
```

```
return 0;
}
```

```
return 0;
}
```

Q21

Test Case

Input

```
10
3 7 5 1 2 9 8 5 3 2
3
```

Output

```
7 7 5 9 9 9 8 5
```

Weightage - 5

Input

```
10
1 2 3 4 5 6 7 8 9 10
5
```

Output

```
5 6 7 8 9 10
```

Weightage - 5

Input

```
42
19 0 30 40 62 7 7 80 95 66 13 95 52 78 66 99 24 28 20 11 5
39
```

Output

```
99 99 99 99
```

Weightage - 10

Input

```
58
93 87 65 1 74 6 98 24 95 0 63 46 4 16 13 13 33 11 4 39 97
52
```

Output

```
98 98 98 98 98 98 98
```

Weightage - 10

Input

```
30
19 59 82 52 47 22 88 31 9 22 66 89 70 18 21 1 4 14 69 41 8
22
```

Output

```
89 91 96 96 96 96 96 96 96
```

Weightage - 10

Input

```
12
33 86 39 55 87 49 66 75 58 17 37 13
9
```

Output

```
87 87 87 87
```

Weightage - 10

Input

```
168
893 639 144 556 370 698 29 659 333 944 872 843 304 883 543 3
131
```

Output

```
995 995 995 995 995 995 995 995 995 995 995 995 995 995 995 9
```

Weightage - 10

Input

```
245
383 113 882 356 2 856 733 419 651 39 293 872 708 782 99 215
112
```

Output

```
990 990 990 990 990 990 990 990 990 990 990 990 990 990 990 9
```

Weightage - 10

Input

Output

-

572 981 240 269 662 146 400 892 920 59 544 147 942 245 146 532 7 286	996 996 996 996 996 996 996 996 996 996 996 996 996 996 996 9

Weightage - 10

Input

Output

572 981 240 269 662 146 400 892 920 59 544 147 942 245 146 532 7 286	996 996 996 996 996 996 996 996 996 996 996 996 996 996 996 9

Weightage - 10

Input

Output

663 208 572 683 383 964 818 94 662 656 445 15 501 48 234 831 741 549	999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 9

Weightage - 10

Sample Input

Sample Output

8 1 3 5 2 1 8 6 9 3	5 5 5 8 8 9

Solution

```
#include<stdio.h>
#include<limits.h>
int main()
{
   int size , arr[1000],ws,ctr,max = INT_MIN,ctr1;
   scanf("%d",&size);
   for( ctr = 0 ; ctr< size ; ctr++)
      scanf("%d ",&arr[ctr]);
  scanf("%d",&ws);
  for( ctr = 0 ; ctr<= size - ws ; ctr++)
     {
     for( ctr1 = 0 ; ctr1 < ws ; ctr1++)
     {
          if( max < arr[ctr+ctr1])
         max = arr[ctr1+ctr];
         }
     printf("%d ",max);
     max = INT_MIN;
        }
   }
```

```
#include<stdio.h>
#include<limits.h>
int main()
{
   int size , arr[1000],ws,ctr,max = INT_MIN,ctr1;
   scanf("%d",&size);
   for( ctr = 0 ; ctr< size ; ctr++)
      scanf("%d ",&arr[ctr]);
  scanf("%d",&ws);
  for( ctr = 0 ; ctr<= size - ws ; ctr++)
     {
     for( ctr1 = 0 ; ctr1 < ws ; ctr1++)
     {
          if( max < arr[ctr+ctr1])
         max = arr[ctr1+ctr];
         }
     printf("%d ",max);
     max = INT_MIN;
        }
   }
```

Q22

Test Case

Input

Output

Malayalam is my mother tongue	is my mother tongue

Weightage - 10

Input

Output

MOM AND DAD ARE MY BEST FRIENDS	AND ARE MY BEST FRIENDS

Weightage - 10

Input

Output

The Current Radar map shows areas of current precipitation at	The Current map shows areas of current precipitation at the na

Weightage - 10

Input	Output
Track rain, snow and storms in Sacramento and Northern Califor	Track rain, snow and storms in Sacramento and Northern Califor

Weightage - 10

Input	Output
Excuse me madam Anna's father is calling anna	Excuse me Anna's father is calling

Weightage - 10

Input	Output
Mom ! let me go out . i want to meet anna	let me go out want to meet

Weightage - 10

Input	Output
Madam ! Anna is hiding a book	is hiding book

Weightage - 10

Input	Output
I am going to Anna house with my mom and dad	am going to house with my and

Weightage - 10

Input	Output
Madam does not know malayalam	does not know

Weightage - 10

Input	Output
I wont watch malayalam films often	wont watch films often

Weightage - 10

Sample Input	Sample Output
He did a good deed	He good

Solution

```
#include<stdio.h>
```

```
#include<stdio.h>
//#include<conio.h>
#include<string.h>

int isPalindrome(char *str)
{
int st,ed;
st = 0;ed = strlen(str)-1;
      while( st < ed)
      {
      if( str[st] == str[ed] || str[st]-32 == str[ed] || str[st]+32 == str[ed])
      {
      st++;
       ed--;
      }
     if( str[st] != str[ed]) return 0;



      }

      return 1;
}

int main()
{
   char str[1000];
 int ctr,st;
// clrscr();
 scanf("%[^\n]s",str);

 for( ctr =0 ,st=0 ; str[ctr] != '\0' ; ctr++ )
 {
   if( str[ctr] == ' ')
   {
     str[ctr]='\0';
     if( isPalindrome(str+st) == 1)
     {
   strcpy( str+st,str+ctr+1);
   ctr = st;

     }
     else
     {
    st = ctr+1;
    str[ctr]=' ';
     }

   }

 }
 if( str[ctr] == '\0' )
 {
    if( isPalindrome(str+st)== 1)
  str[st-1]='\0';
   }
   printf("%s",str);
}



#include<stdio.h>
//#include<conio.h>
#include<string.h>

int isPalindrome(char *str)
{
int st,ed;
st = 0;ed = strlen(str)-1;
      while( st < ed)
      {
      if( str[st] == str[ed] || str[st]-32 == str[ed] || str[st]+32 == str[ed])
      {
      st++;
       ed--;
      }
     if( str[st] != str[ed]) return 0;



      }
```

```
      return 1;
}

int main()
{
   char str[1000];
   int ctr,st;
  // clrscr();
   scanf("%[^\n]s",str);

   for( ctr =0 ,st=0 ; str[ctr] != '\0' ; ctr++ )
   {
     if( str[ctr] == ' ')
     {
       str[ctr]='\0';
       if( isPalindrome(str+st) == 1)
       {
     strcpy( str+st,str+ctr+1);
     ctr = st;

       }
       else
       {
      st = ctr+1;
      str[ctr]=' ';
       }
      }

    }
    if( str[ctr] == '\0' )
    {
       if( isPalindrome(str+st)== 1)
     str[st-1]='\0';
    }
    printf("%s",str);
 }
```

Q23

Test Case

Input

```
49
823 923 227 156 105 899 833 708 350 184 833 827 685 781 404 
```

Output

```
420 864 756 920 832 156 708 350 156 693 105 184 530 664 470 
```

Weightage - 10

Input

```
109
791 206 106 248 137 527 792 401 398 305 155 927 716 273 989 
```

Output

```
792 864 624 468 858 966 920 616 220 928 495 364 968 208 676 
```

Weightage - 10

Input

```
586
115 611 259 490 612 624 519 479 47 436 884 366 660 343 885 8
```

Output

```
840 840 960 960 660 420 864 990 924 990 990 780 630 624 880 
```

Weightage - 10

Input

```
952
340 165 564 289 494 443 644 120 859 506 722 473 193 881 396 
```

Output

```
840 720 720 960 660 780 672 540 924 936 432 528 240 912 912 
```

Weightage - 10

Input

Output

```
924
404 289 842 357 808 973 329 774 86 739 883 376 631 517 678 44
```

```
960 960 756 672 756 756 936 936 780 672 660 420 360 504 360
```

Weightage - 10

Input

Output

```
518
523 586 850 230 763 226 243 33 587 176 142 674 658 366 982 8
```

```
840 960 480 780 504 864 864 576 336 816 240 588 800 450 468
```

Weightage - 10

Input

Output

```
69
658 519 610 570 630 189 299 413 467 770 997 401 595 348 259
```

```
630 972 252 570 770 616 408 696 750 510 784 348 380 72 774 5
```

Weightage - 10

Input

Output

```
118
737 681 545 340 64 814 900 325 615 353 556 899 104 50 48 782
```

```
900 540 396 252 910 920 330 552 888 400 324 340 732 726 996
```

Weightage - 10

Input

Output

```
538
358 356 299 425 112 131 564 675 702 652 916 154 858 264 320
```

```
900 660 756 660 864 630 480 672 540 864 504 936 660 624 528
```

Weightage - 10

Input

Output

```
36
10 8 17 8 43 12 37 39 12 37 31 1 35 10 11 3 12 33 41 27 48
```

```
48 30 40 12 12 12 18 16 10 8 8 39 35 10 33 27 39 35 26 46 2
```

Weightage - 5

Input

Output

```
32
9 30 11 30 2 28 4 8 43 49 2 17 46 46 9 20 8 32 44 26 3 45
```

```
30 30 28 20 32 44 45 8 46 46 8 26 10 39 46 46 34 6 9 4 49
```

Weightage - 5

Sample Input

Sample Output

```
5
8 2 3 12 16
```

```
12 16 8 2 3
```

Solution

```
#include<stdio.h>
#include<math.h>
int findFactCount( int n)
{
    int ctr,sqr,count=2;
    sqr = (int)sqrt(n);
    if(n == 1)
    count--;
    else
    {
    for( ctr = 2 ; ctr<= sqr ; ctr++ )
```


```
#include<stdio.h>
#include<math.h>
int findFactCount( int n)
{
    int ctr,sqr,count=2;
    sqr = (int)sqrt(n);
    if(n == 1)
    count--;
    else
    {
    for( ctr = 2 ; ctr<= sqr ; ctr++ )
```

```
      if( n % ctr == 0 )

          count +=2;
    ctr--;
    if( ctr * ctr == n )
       count--;

    }
    return count;
}
int main()
{
    int arr[1000],fact[1000],ctr,size,max=-1,maxInd=-1,temp,ctr1;
    scanf("%d",&size);
    for( ctr =0 ; ctr < size ; ctr++ )
       scanf("%d",&arr[ctr]);
    for( ctr =0 ; ctr < size ; ctr++ )
           fact[ctr]=findFactCount(arr[ctr]);
    for( ctr =0 ; ctr < size ; ctr++ )
    {
           for( ctr1 =0 ; ctr1 < size ; ctr1++ )
           {
               if( fact[ctr1] != -1 && max < fact[ctr1])
               {
                      max = fact[ctr1];
                      maxInd = ctr1;
               }
              /* temp = arr[ctr];
               arr[ctr] = arr[maxInd];
               arr[maxInd] = temp;

               temp = fact[ctr];
               fact[ctr] = fact[maxInd];
               fact[maxInd] = temp;*/
               fact[maxInd] = -1;
                          printf("%d ",arr[maxInd]);
               max = -1;
               maxInd=-1;
    }

    /*for( ctr =0 ; ctr < size ; ctr++ )
       printf("%d ",arr[ctr]);
*/
}
```

```
      if( n % ctr == 0 )

          count +=2;
    ctr--;
    if( ctr * ctr == n )
       count--;

    }
    return count;
}
int main()
{
    int arr[1000],fact[1000],ctr,size,max=-1,maxInd=-1,temp,ctr1;
    scanf("%d",&size);
    for( ctr =0 ; ctr < size ; ctr++ )
       scanf("%d",&arr[ctr]);
    for( ctr =0 ; ctr < size ; ctr++ )
           fact[ctr]=findFactCount(arr[ctr]);
    for( ctr =0 ; ctr < size ; ctr++ )
    {
           for( ctr1 =0 ; ctr1 < size ; ctr1++ )
           {
               if( fact[ctr1] != -1 && max < fact[ctr1])
               {
                      max = fact[ctr1];
                      maxInd = ctr1;
               }
              /* temp = arr[ctr];
               arr[ctr] = arr[maxInd];
               arr[maxInd] = temp;

               temp = fact[ctr];
               fact[ctr] = fact[maxInd];
               fact[maxInd] = temp;*/
               fact[maxInd] = -1;
                          printf("%d ",arr[maxInd]);
               max = -1;
               maxInd=-1;
    }

    /*for( ctr =0 ; ctr < size ; ctr++ )
       printf("%d ",arr[ctr]);
*/
}
```

Q24

Test Case

Input

```
10 50
```

Output

```
11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
```

Weightage - 10

Input

```
100 525
```

Output

```
101 103 105 107 109 111 113 115 117 119 121 123 125 127 129
```

Weightage - 10

Input

```
1000 1256
```

Output

```
1001 1003 1005 1007 1009 1011 1013 1015 1017 1019 1021 1023 1
```

Weightage - 10

Input

```
555 1000
```

Output

```
555 557 559 561 563 565 567 569 571 573 575 577 579 581 5
```

Weightage - 10

Input

```
100 1000
```

Output

```
101 103 105 107 109 111 113 115 117 119 121 123 125 127 129
```

Weightage - 10

Input

```
1500 5000
```

Output

```
1501 1503 1505 1507 1509 1511 1513 1515 1517 1519 1521 1523 1
```

Weightage - 10

Input

```
5000 8000
```

Output

```
5001 5003 5005 5007 5009 5011 5013 5015 5017 5019 5021 5023 5
```

Weightage - 10

Input

```
8235 9315
```

Output

```
8235 8237 8239 8241 8243 8245 8247 8249 8251 8253 8255 8257 8
```

Weightage - 10

Input

```
5 6
```

Output

```
5
```

Weightage - 5

Input

```
225 235
```

Output

```
225 227 229 231 233
```

Weightage - 5

Input

```
89 125
```

Output

```
89 91 93 95 97 99 101 103 105 107 109 111 113 115 117 119 12
```

Weightage - 10

Sample Input

```
2 15
```

Sample Output

```
3 5 7 9 11 13
```

Solution

```c
#include<stdio.h>

int main()
{
    int start , end,ctr;
```

```
    scanf("%d %d",&start,&end);
    if( start % 2 == 0 )
       start++;
    for( ctr = start ; ctr < end ; ctr+=2 )
         printf("%d ", ctr);
   return 0;
}
```

Q25

Test Case

Input

```
(a+b+c)[(d+e)(f*g)]
```

Output

```
VALID
```

Weightage - 10

Input

```
(a+b))(c*d)
```

Output

```
INVALID
```

Weightage - 10

Input

```
(a+b/c-e)[a-b]
```

Output

```
VALID
```

Weightage - 20

Input

```
(((a+b+c+d+e+f+g+h)))
```

Output

```
VALID
```

Weightage - 10

Input

```
((ab+))
```

Output

```
INVALID
```

Weightage - 20

Input

```
(a+b)(c+d)(e+f)(a+b)(d+y+k+o)
```

Output

```
VALID
```

Weightage - 20

Input

```
(abdjcbc)
```

Output

```
INVALID
```

Weightage - 10

Sample Input

```
(a+b)(c+d+e)
```

Sample Output

```
VALID
```

Sample Input

```
(a+b)(c+d)
```

Sample Output

```
VALID
```

Sample Input

```
(a+b))
```

Sample Output

```
INVALID
```

Sample Input

```
(ab+)
```

Sample Output

```
INVALID
```

Solution

```
#include<stdio.h>
int strChr(char *str , char ch)
{
    int ind;
    for(ind = 0 ; str[ind] ; ind++)
    {
        if(str[ind] == ch)
            return 1;
    }
return 0;
}
int main()
{
char str[100] , pop;
char close[5] = ")]" , open[5]="([" ,operators[6] ="+*/-";
scanf("%s" , str);
char paran[100] , operand[100];
int top_p , top_o , ind , flag ;
top_p = top_o = -1;

   for(ind = 0 ; str[ind] ; ind++)
   {
       if(strChr(open , str[ind]) == 1)
          paran[++top_p] = str[ind];
       else if(strChr(close , str[ind]) == 1 )
       {
           if(top_p == -1)
              break;
           else
           {
               pop = paran[top_p];
               flag = 0 ;
               switch(pop)
           {
               case '(' : if(str[ind] != ')') flag = 1 ; break;
               case '[' : if(str[ind] != ']') flag = 1 ; break;
               case '{' : if(str[ind] != '}') flag = 1 ; break;
           }
           if(flag == 1) break;
           else
               paran[top_p--] = 0;
    }
}
else if(  str[ind] >= 'a' && str[ind] <= 'z' )
   operand[++top_o] = str[ind];
else if( strChr(operators , str[ind]) == 1)
{
    if( top_o == -1) break;
    else
    {
        operand[top_o--] = 0;
        if( !(str[ind+1] >= 'a' && str[ind+1] <= 'z'))
           break;
        else
        {
         ind++;
         if(strChr(operators , str[ind+1]) == 1)
             operand[++top_o] = str[ind];

        }
```

```
            }
        }
    }
if(str[ind] == 0 && top_o == -1 && top_p == -1)
     printf("VALID");
else
    printf("INVALID");

return 0 ;
}



#include<stdio.h>
int strChr(char *str , char ch)
{
    int ind;
    for(ind = 0 ; str[ind] ; ind++)
    {
        if(str[ind] == ch)
             return 1;
    }
return 0;
}
int main()
{
char str[100] , pop;
char close[5] = ")]" , open[5]="([" ,operators[6] ="+*/-";
scanf("%s" , str);
char paran[100] , operand[100];
int top_p , top_o , ind , flag ;
top_p = top_o = -1;

   for(ind = 0 ; str[ind] ; ind++)
   {
       if(strChr(open , str[ind]) == 1)
          paran[++top_p] = str[ind];
       else if(strChr(close , str[ind]) == 1 )
       {
           if(top_p == -1)
              break;
           else
           {
               pop = paran[top_p];
               flag = 0 ;
               switch(pop)
               {
                   case '(' : if(str[ind] != ')') flag = 1 ; break;
                   case '[' : if(str[ind] != ']') flag = 1 ; break;
                   case '{' : if(str[ind] != '}') flag = 1 ; break;
               }
               if(flag == 1) break;
               else
                    paran[top_p--] = 0;
           }
       }
       else if(  str[ind] >= 'a' && str[ind] <= 'z' )
          operand[++top_o] = str[ind];
       else if( strChr(operators , str[ind]) == 1)
       {
           if( top_o == -1) break;
           else
           {
               operand[top_o--] = 0;
               if( !(str[ind+1] >= 'a' && str[ind+1] <= 'z'))
                  break;
               else
               {
                ind++;
                if(strChr(operators , str[ind+1]) == 1)
                    operand[++top_o] = str[ind];
               }


           }
       }
   }
if(str[ind] == 0 && top_o == -1 && top_p == -1)

     printf("VALID");
```

```
else
    printf("INVALID");

return 0 ;
}
```

Q26

Test Case

Input

```
1987
```

Output

```
Non-leap
```

Weightage - 10

Input

```
2020
```

Output

```
Leap
```

Weightage - 10

Input

```
2400
```

Output

```
Leap
```

Weightage - 10

Input

```
2345
```

Output

```
Non-leap
```

Weightage - 10

Input

```
3400
```

Output

```
Non-leap
```

Weightage - 10

Input

```
3600
```

Output

```
Leap
```

Weightage - 10

Input

```
2478
```

Output

```
Non-leap
```

Weightage - 10

Input

```
4560
```

Output

```
Leap
```

Weightage - 10

Input

```
6747
```

Output

```
Non-leap
```

Weightage - 10

Input

```
6727
```

Output

```
Non-leap
```

Weightage - 10

Sample Input

```
1990
```

Sample Output

```
Non-leap
```

Sample Input

```
2000
```

Sample Output

```
Leap
```

Solution

```
#include<stdio.h>
int main()
{

int year;
scanf("%d" , &year);
if(year % 4 == 0 && year % 100 != 0 || year % 400 == 0)
     printf("Leap");
else
    printf("Non-leap");

return 0;
}
```

```
#include<stdio.h>
int main()
{

int year;
scanf("%d" , &year);
if(year % 4 == 0 && year % 100 != 0 || year % 400 == 0)
     printf("Leap");
else
    printf("Non-leap");

return 0;
}
```

Q27

Test Case

Input

```
50
184 87 178 116 194 136 187 93 50 22 163 28 91 60 164 127 141
```

Output

```
184-1
87-1
178-1
116-2
```

Weightage - 10

Input

```
150
184 87 178 116 194 136 187 93 50 22 163 28 91 60 164 127 141
```

Output

```
184-1
87-2
178-1
116-3
```

Weightage - 20

Input

```
75
1384 887 778 916 1794 336 1387 493 650 1422 363 28 691 60 176
```

Output

```
1384-1
887-1
778-1
916-1
```

Weightage - 10

Input

```
121
4 7 8 6 4 6 7 3 10 2 3 8 1 10 4 7 1 7 3 7 2 9 8 10 3 1 3
```

Output

```
4-11
7-17
8-13
6-11
```

Weightage - 20

Input

```
500
4 7 8 6 4 6 7 3 10 2 3 8 1 10 4 7 1 7 3 7 2 9 8 10 3 1 3
```

Output

```
4-47
7-54
8-44
6-42
```

Weightage - 20

Input

```
1500
4 7 8 6 4 6 7 3 10 2 3 8 1 10 4 7 1 7 3 7 2 9 8 10 3 1 3
```

Output

```
4-157
7-144
8-148
6-134
```

Weightage - 10

Input

```
10
884 887 778 416 294 836 887 493 1150 422
```

Output

```
884-1
887-2
778-1
416-1
```

Weightage - 5

Input

```
10
4 7 18 16 14 16 7 13 10 2
```

Output

```
4-1
7-2
18-1
16-2
```

Weightage - 5

Sample Input

```
10
4 7 18 16 14 16 7 13 10 2
```

Sample Output

```
4-1
7-2
18-1
16-2
```

Solution

```
#include<stdio.h>
#include<malloc.h>
int main()
{

int N , ind , num , counter , flag , ind1;
scanf("%d" , &N);
int *arr , *count;
arr= (int*)calloc(N , sizeof(int));
count= (int*)calloc(N , sizeof(int));
counter = 0;
for(ind = 0 ; ind < N ; ind++)
{
    scanf("%d" ,&num);
    for(ind1 = 0, flag = 0 ; ind1 < counter ; ind1++)
    {
        if(arr[ind1] == num)
        {
            count[ind1]++;
            flag = 1;
        }

    }
    if(flag == 0)
    {
        arr[counter] = num;
        count[counter]++;
```


```
#include<stdio.h>
#include<malloc.h>
int main()
{

int N , ind , num , counter , flag , ind1;
scanf("%d" , &N);
int *arr , *count;
arr= (int*)calloc(N , sizeof(int));
count= (int*)calloc(N , sizeof(int));
counter = 0;
for(ind = 0 ; ind < N ; ind++)
{
    scanf("%d" ,&num);
    for(ind1 = 0, flag = 0 ; ind1 < counter ; ind1++)
    {
        if(arr[ind1] == num)
        {
            count[ind1]++;
            flag = 1;
        }

    }
    if(flag == 0)
    {
        arr[counter] = num;
        count[counter]++;
```

```
        counter++;

    }
}
for(ind = 0 ; ind < counter ; ind++)
    printf("%d-%d\n" , arr[ind] , count[ind]);

  return 0;
}
```

```
        counter++;

    }
}
for(ind = 0 ; ind < counter ; ind++)
    printf("%d-%d\n" , arr[ind] , count[ind]);

  return 0;
}
```

Q28

Test Case

Input

```
Fiih+!\,ln
```

Output

```
nlhi+!\,iF
```

Weightage - 10

Input

```
Fiih+!\,ln:nRZfHJZd'<hFRcuBvXoQThprY n;v7oOsolrX!e
```

Output

```
eXrl+!\,os:Oo7vnYrp'<hTQoXvBucRFhdZJ H;fZRnnlhii!F
```

Weightage - 10

Input

```
Fiih+!\,ln:nRZfHJZd'<hFRcuBvXoQThprY n;v7oOsolrX!e6W9f9;F1PT?08FW
```

Output

```
zaUp+!\,WF:80TP1F9f'<9W6eXrlosOo7vnY r;phTQoXvBu!cRFhdZ;JHfZ?Rnnl
```

Weightage - 10

Input

```
Fiih+!\,ln:nRZfHJZd'
```

Output

```
dZJH+!\,fZ:RnnlhiiF'
```

Weightage - 10

Input

```
Fiih+!\,ln:nRZfHJZd'<hFRcuBvXoQThprY n;v7oOsolrX!e6W9f9;F1PT?08FW
```

Output

```
PB2h+!\,tN:hDeika0k'<ga6OaCImqno805S K;74T3TmCtz!dffoFD;RUUM?oaaS
```

Weightage - 20

Input

```
Fiih+!\,ln:nRZfHJZd'<hFRcuBvXo
```

Output

```
oXvB+!\,uc:RFhdZJHf'<ZRnnlhiiF
```

Weightage - 10

Input

```
Fiih+!\,ln:nRZfHJZd'<hFRcuBvXoQThprY n;v7oOsolrX!e6W9f9;F1PT?08FW
```

Output

```
kjFQ+!\,b4:5x0IsdPB'<2htNhDeika0kga6 O;aCImqno80!5SK74T;3TmC?tzdf
```

Weightage - 10

Input

```
Fiih+!\,ln:nRZfHJZd'<hFRcuBvXoQThprY n;v7oOsolrX!e6W9f9
```

Output

```
9f9W+!\,6e:XrlosOo7'<vnYrphTQoXvBucR F;hdZJHfZRn!nlhiiF
```

Weightage - 10

Input

```
Fiih+!\,ln:nRZfHJZd'<hFRc
```

Output

```
cRFh+!\,dZ:JHfZRnnl'<hiiF
```

Weightage - 10

Sample Input

```
A man, in the boat says : I see 1-2-3 in the sky
```

Sample Output

```
y kse, ht ni3 21ee sIsy : a sta o-b-e ht nin amA
```

Solution

```
#include<stdio.h>
#define isAllow(ch) (ch >='0' && ch <= '9' || ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z' )
int main()
{
char str[200] , temp;
scanf("%[^\n]s" , str);
int start , end ;

for(end = 0 ; str[end] ; end++);
end--;
start = 0;
  while(start < end)
  {
      while( str[start] && !(isAllow(str[start]))) start++;
      while( str[end] && !(isAllow(str[end]))) end--;
      if(start < end)
      {
          temp = str[start];
          str[start] = str[end];
          str[end] = temp;
          start++;
          end--;
      }
  }

printf("%s " , str);
return 0;
}
```

Q29

Test Case

Input

```
PROGRAM
```

Output

```
   G
  GR
 GRA
GRAM
```

Weightage - 20

Input

```
WONDERS
```

Output

```
   D
  DE
 DER
DERS
```

Weightage - 10

Input

```
LOVELYO
```

Output

```
   E
  EL
 ELY
ELYO
```

Weightage - 10

Input

```
12345
```

Output

```
   3
  34
 345
3451
```

Weightage - 10

Input

```
ONETWOTHREE
```

Output

```
     O
    OT
   OTH
OTHR
```

Weightage - 10

Input

```
TWO
```

Output

```
  W
 WO
WOT
```

Weightage - 10

Input

```
ELEVENN
```

Output

```
    V
   VE
  VEN
VENN
```

Weightage - 10

Input

```
LOVEINDIA
```

Output

```
     I
    IN
   IND
INDI
```

Weightage - 10

Input

```
ABCDEFGHI
```

Output

```
     E
    EF
   EFG
EFGH
```

Weightage - 10

Sample Input

```
Hello
```

Sample Output

```
  l
 ll
llo
lloH
```

Solution

```
#include<stdio.h>
int main()
{
char str[100];
scanf("%s" , str);
int len ,row , col , mid , space;

for(len = 0 ; str[len] ; len++);
space = len - 1;
for(row = 1 ; row <= len/2+1 ; row++,space-- ,printf("\n"))
{
    for(col = 1 ; col <= space ; printf(" "), col++);
    for(col = 1 , mid = len/2 ; col <= (len - space) ; col++)
         printf("%c", str[mid++]);
}

//space+=2;
```

```
#include<stdio.h>
int main()
{
char str[100];
scanf("%s" , str);
int len ,row , col , mid , space;

for(len = 0 ; str[len] ; len++);
space = len - 1;
for(row = 1 ; row <= len/2+1 ; row++,space-- ,printf("\n"))
{
    for(col = 1 ; col <= space ; printf(" "), col++);
    for(col = 1 , mid = len/2 ; col <= (len - space) ; col++)
         printf("%c", str[mid++]);
}

//space+=2;
```

```
for(row = 1 ; row <= len/2 ; row++,space-- ,printf("\n"))
{
    for(col = 1 ; col <= space ; printf(" "), col++);
    for(mid = len/2 ; str[mid]; mid++)
         printf("%c", str[mid]);
    for(col = 1 ; col <= row ; col++)
        printf("%c" , str[col-1]);
}

  return 0;
}
```

```
for(row = 1 ; row <= len/2 ; row++,space-- ,printf("\n"))
{
    for(col = 1 ; col <= space ; printf(" "), col++);
    for(mid = len/2 ; str[mid]; mid++)
         printf("%c", str[mid]);
    for(col = 1 ; col <= row ; col++)
        printf("%c" , str[col-1]);
}

  return 0;
}
```

Q30

Test Case

Input

```
one two three
```

Output

```
three two one
```

Weightage - 10

Input

```
sdjgfvjdsv dfvgjdfvj jgv khjkfde yusfd ijkhfv
```

Output

```
ijkhfv yusfd khjkfde jgv dfvgjdfvj sdjgfvjdsv
```

Weightage - 10

Input

```
snbdcvdnsc dvcgdvc dhcvhgdvc hdgvchjvc hdgchjdcv hdcvhvc hsdvch
```

Output

```
haai hjdchj hdbc hjdchvhj jdcbghj hdc hsdvch hdcvhvc hdgchjdcv
```

Weightage - 10

Input

```
one two three four five six
```

Output

```
six five four three two one
```

Weightage - 10

Input

```
india delhi mumbai sivakasi virudhunagar cbe chennai ooty
```

Output

```
ooty chennai cbe virudhunagar sivakasi mumbai delhi india
```

Weightage - 10

Input

```
dxhcbndcv nd hjcv ysv hsvhjvssvsvsv cjhvcsv cjvjvcjd vcjvdjcv j
```

Output

```
xcjvwadfuwdsyu adxiuahxkbsc djcvhjvchjvd vcjvdchjv jvcjvdcvdvchj
```

Weightage - 10

Input

```
zxnb djbc djcvhjdvc hjdgc hdcg jdcg djcg uidgci dgci diu uidcg
```

Output

```
d chgibcdc uidcgiugdc diu dgci uidgci djcg jdcg hdcg hjdgc djc
```

Weightage - 20

Input

Output

sdhgcjh jhcv shjcvhjcv	shjcvhjcv jhcv sdhgcjh

Weightage - 10

Input **Output**

sjchb cbh dchjb	dchjb cbh sjchb

Weightage - 10

Sample Input **Sample Output**

i love india	india love i

Solution

```
#include<stdio.h>
char * strReverse(char * str)
{
    int start , end ;
    char temp;
    for(end = 0 ; str[end] ; end++);
    for(start = 0 , --end ; start < end ; start++ , end--)
    {
        temp = str[start];
        str[start] = str[end];
        str[end] = temp;
    }
return str;
}
char * wordReversal(char * str , char* space)
{
    int ind;
   if(space[0] == 0)
      return strReverse(str);
   for(ind = (space - str)+1 ; str[ind] && str[ind] != 32 ; ind++);
   wordReversal(space+1 , str+ind);
   *space = 0;
   strReverse(str);
   *space = 32;
return str;
}

int main()
{
char str[200];
int ind;
scanf("%[^\n]s" , str);
strReverse(str);
for(ind = 0 ; str[ind] && str[ind] != 32 ; ind++);
wordReversal(str , str+ind);
printf("%s" , str);



return 0;
}



#include<stdio.h>
char * strReverse(char * str)
{
    int start , end ;
    char temp;
    for(end = 0 ; str[end] ; end++);
    for(start = 0 , --end ; start < end ; start++ , end--)
    {
        temp = str[start];
        str[start] = str[end];
        str[end] = temp;
    }
```

```
return str;
}
char * wordReversal(char * str , char* space)
{
    int ind;
   if(space[0] == 0)
      return strReverse(str);
   for(ind = (space - str)+1 ; str[ind] && str[ind] != 32 ; ind++);
   wordReversal(space+1 , str+ind);
   *space = 0;
   strReverse(str);
   *space = 32;
return str;
}

int main()
{
char str[200];
int ind;
scanf("%[^\n]s" , str);
strReverse(str);
for(ind = 0 ; str[ind] && str[ind] != 32 ; ind++);
wordReversal(str , str+ind);
printf("%s" , str);
return 0;
}
```

Q31

Test Case

Input

```
123456789 678912345
```

Output

```
Yes
```

Weightage - 10

Input

```
123456789 679812345
```

Output

```
No
```

Weightage - 10

Input

```
1234678 6781234
```

Output

```
Yes
```

Weightage - 10

Input

```
1234678 6782134
```

Output

```
No
```

Weightage - 10

Input

```
12345 12345
```

Output

```
Yes
```

Weightage - 10

Input

```
12345 67892
```

Output

```
No
```

12345 67892	No

Weightage - 20

Input	Output
23456 62345	Yes

Weightage - 10

Input	Output
2342432424 4234243242	Yes

Weightage - 10

Input	Output
456 564	Yes

Weightage - 10

Sample Input	Sample Output
12345 45123	Yes

Sample Input	Sample Output
12345 54123	No

Solution

```
#include<stdio.h>
int main()
{
long long int num1 , num2 , power , nod , rotate;
scanf("%lld%lld",&num1 , &num2);
power = 1;
nod = 0;
while(num1 / power)
{
    power *= 10;
    nod++;
}
power /= 10;

rotate = num1;
while(rotate != num2 && nod)
{
    rotate = (rotate%10)*power + (rotate/10);
    nod--;
}
if(rotate == num2 && nod != 0 )
     printf("Yes");
else
    printf("No");

return 0;
}
```

```
#include<stdio.h>
int main()
{
long long int num1 , num2 , power , nod , rotate;
scanf("%lld%lld",&num1 , &num2);
power = 1;
nod = 0;
while(num1 / power)
{
    power *= 10;
    nod++;
}
power /= 10;

rotate = num1;
while(rotate != num2 && nod)
{
    rotate = (rotate%10)*power + (rotate/10);
    nod--;
}
if(rotate == num2 && nod != 0 )
     printf("Yes");
else
    printf("No");

return 0;
}
```

Q32

Test Case

Input

6

Output

666666
655556
654456
654456

Weightage - 20

Input

2

Output

22
22

Weightage - 10

Input

5

Output

5555
5445
5445
5555

Weightage - 10

Input

7

Output

777777
766667
765567
765567

Weightage - 20

Input

8

Output

88888888
87777778
87666678
87655678

Weightage - 10

Input

9

Output

99999999
98888889
98777789
98766789

Weightage - 20

Input

3

Output

33
33

Weightage - 10

Sample Input

4

Sample Output

4444
4334
4334
4444

Solution

```
#include<stdio.h>
int setNum(int N , int *pow)
{
    int ind , num;
    num = 0 ;
    *pow = 1;
    for(ind = 1 ; ind <= N/2 ; num = num * 10 + N , (*pow) = (*pow) * 10 + 1 , ind++);
    (*pow) /= 100;
return num;
```

```
}
int reverse(int num)
{
    int revnum = 0;
    while(num)
    {
        revnum =  revnum * 10 + (num % 10);
        num /= 10;
    }
return revnum;
}
int main()
{
int N  , num , pow , row , col;
scanf("%d" , &N);
num = setNum(N , &pow);
for(row = 1 ; row <= N/2 ; row++,printf("\n"))
{
    printf("%d%d",num,reverse(num));
    num = num - pow;
    pow /= 10;
}
for(row = 1 ; row <= N/2 ; row++,printf("\n"))
{
    printf("%d%d",num,reverse(num));
    pow = pow * 10 + 1;
    num = num + pow;
}


return 0;
}
```

```
#include<stdio.h>
int setNum(int N , int *pow)
{
    int ind , num;
    num = 0 ;
    *pow = 1;
    for(ind = 1 ; ind <= N/2 ; num = num * 10 + N , (*pow) = (*pow) * 10 + 1 , ind++);
    (*pow) /= 100;
return num;
}
int reverse(int num)
{
    int revnum = 0;
    while(num)
    {
        revnum =  revnum * 10 + (num % 10);
        num /= 10;
    }
return revnum;
}
int main()
{
int N  , num , pow , row , col;
scanf("%d" , &N);
num = setNum(N , &pow);
for(row = 1 ; row <= N/2 ; row++,printf("\n"))
{
    printf("%d%d",num,reverse(num));
    num = num - pow;
    pow /= 10;
}
for(row = 1 ; row <= N/2 ; row++,printf("\n"))
{
    printf("%d%d",num,reverse(num));
    pow = pow * 10 + 1;
    num = num + pow;
}


return 0;
}
```

Q33

Test Case

Input

```
1234
```

Output

```
5555
```

Weightage - 10

Input

```
354656
```

Output

```
11244211
```

Weightage - 10

Input

```
3656
```

Output

```
125521
```

Weightage - 10

Input

```
24546
```

Output

```
39633693
```

Weightage - 10

Input

```
1234567
```

Output

```
8888888
```

Weightage - 10

Input

```
3545
```

Output

```
8998
```

Weightage - 10

Input

```
23
```

Output

```
55
```

Weightage - 10

Input

```
5670
```

Output

```
59895
```

Weightage - 20

Input

```
34
```

Output

```
77
```

Weightage - 10

Sample Input

```
32
```

Sample Output

```
55
```

Solution

```
#include<stdio.h>
long long int reverse(long long int num)
{
    long long int revnum = 0;
    while(num)
    {
        revnum = revnum * 10 +(num%10);
        num /= 10;
    }
return revnum;
}
int main()
{
long long int num , sum ;
scanf("%lld" , &num);
      while(1)
      {
        sum = num + reverse(num);
        if( sum == reverse(sum))
          break;
        num = sum;
      }
      printf("%lld" , sum);
return 0;
}
```

```
#include<stdio.h>
long long int reverse(long long int num)
{
    long long int revnum = 0;
    while(num)
    {
        revnum = revnum * 10 +(num%10);
        num /= 10;
    }
return revnum;
}
int main()
{
long long int num , sum ;
scanf("%lld" , &num);
      while(1)
      {
        sum = num + reverse(num);
        if( sum == reverse(sum))
          break;
        num = sum;
      }
      printf("%lld" , sum);
return 0;
}
```

Q34

Test Case

Input

```
Fiih+!\,ln:nRZfHJZd'<hFRc
Fiih+!\,ln
```

Output

```
:RZfHJZd'<Rc
```

Weightage - 10

Input

```
Fiih+!\,ln:nRZfHJZd'<hFRcuBvXoQThprY n;v7oOsolrX!e6W9f9;F1PT?08FW
Fiih+!\,ln:nRZfHJZd'<hFRcuBvXo
```

Output

```
iih+,ZHJZ'<hcuBQhY
```

Weightage - 10

Input

```
sdjkhfjdbvjfbvjfbhvjhbfvhjbjbvjhbfrvhjbrfgbvrvhjhjbjbfjbvhjbfhvjb
bvf
```

Output

```
sdjkhjdjjhjhhjjjhrhjrgrhjhjjjhjhjdjhjhj
```

Weightage - 10

Input

```
jfbhvjbfjfbvdjlakjdklafckajbvlasndwklfjkleghbvkjbsbckjlehnvkjrvjel
ak
```

Output

```
jfbhvjbfjfbvdjljdlfcjbvlsndwlfjleghbvjbsbcjlehnvjrvjebfbvjbfje
```

Weightage - 20

Input

```
vhbfhvjbhfvhjbfvjbfbvjd
dbchfn
```

Output

```
vvjvjvjvj
```

Weightage - 20

Input

```
cjhbjdbvkjdbvjkeakfbgejugfyuiegfesugfiu
hdhjfbf
```

Output

```
cvkvkeakgeugyuiegesugiu
```

Weightage - 10

Input

```
dnbvjhbv
dfhvbhjbvb
```

Output

```
n
```

Weightage - 10

Input

```
fdkjvhjkfbvkjbfkjbvfjkbv
abchdb
```

Output

```
fkjvjkfvkjfkjvfjkv
```

Weightage - 10

Sample Input

```
experience
enc
```

Sample Output

```
xpri
```

Solution

```
#include<stdio.h>
int main()
{
char s1[200] , s2[200];
int ind1 , ind2;
scanf("%s%s" , s1 , s2);

for(ind2 = 0 ; s2[ind2] ; ind2++)
{
    for(ind1 = 0 ; s1[ind1] ; ind1++)
    {
        if(s1[ind1] == s2[ind2])
            s1[ind1] = '@';
    }
for(ind1 = 0 ; s1[ind1] && s1[ind1] != '@' ; ind1++);
ind2 = ind1;

while(1)
{
for(ind2; s1[ind2] && s1[ind2] == '@' ; ind2++);
while( s1[ind2] && s1[ind2] != '@')
{
    s1[ind1] = s1[ind2];
    ind1++;
    ind2++;
}
if(s1[ind2] == 0)  break;
}
s1[ind1] = 0;
printf("%s" , s1);
return 0;
}
```

```
#include<stdio.h>
int main()
{
char s1[200] , s2[200];
int ind1 , ind2;
scanf("%s%s" , s1 , s2);

for(ind2 = 0 ; s2[ind2] ; ind2++)
{
    for(ind1 = 0 ; s1[ind1] ; ind1++)
    {
        if(s1[ind1] == s2[ind2])
            s1[ind1] = '@';
    }
for(ind1 = 0 ; s1[ind1] && s1[ind1] != '@' ; ind1++);
ind2 = ind1;

while(1)
{
for(ind2; s1[ind2] && s1[ind2] == '@' ; ind2++);
while( s1[ind2] && s1[ind2] != '@')
{
    s1[ind1] = s1[ind2];
    ind1++;
    ind2++;
}
if(s1[ind2] == 0)  break;
}
s1[ind1] = 0;
printf("%s" , s1);
return 0;
}
```

Q35

Test Case

Input

Output

27	AA

Weightage - 5

Input	Output
987654321	CECGIBQ

Weightage - 10

Input	Output
556	UJ

Weightage - 5

Input	Output
123456789	JJDDJA

Weightage - 10

Input	Output
16031994	AIBCYD

Weightage - 10

Input	Output
24031995	AZOHGM

Weightage - 10

Input	Output
6101965	MIDNY

Weightage - 10

Input	Output
1000	ALL

Weightage - 10

Input	Output
676	YZ

Weightage - 10

Input	Output

16384	XFD

Weightage - 10

Input	Output
27122005	BGICHA

Weightage - 10

Sample Input	Sample Output
26	Z

Solution

```
#include <stdio.h>
#include<string.h>
int main() {
    //code
    unsigned int testnum=101027545,rem,ind,cases,tc,start,end;
    char str[100],temp;
  //  scanf("%u",&cases);
   // for(tc =0 ; tc < cases ; tc++)
   // {
        scanf("%u",&testnum);
    ind=0;
    while(testnum)
    {
        rem = (testnum% 26);
        if(rem==0) {rem=26; testnum--;}
        //printf("%c",rem+64);
        str[ind++]=rem+64;
        testnum/=26;
    }
    str[ind]='\0';
    start=0;
    end=ind-1;
    while(start<end)
    {
        temp = str[start];
        str[start]=str[end];
        str[end]=temp;
        start++;
        end--;
    }
    //strrev(str);
    printf("%s",str);
   // }
    return 0;
}
```

```
#include <stdio.h>
#include<string.h>
int main() {
    //code
    unsigned int testnum=101027545,rem,ind,cases,tc,start,end;
    char str[100],temp;
  //  scanf("%u",&cases);
   // for(tc =0 ; tc < cases ; tc++)
   // {
        scanf("%u",&testnum);
    ind=0;
    while(testnum)
    {
        rem = (testnum% 26);
        if(rem==0) {rem=26; testnum--;}
        //printf("%c",rem+64);
        str[ind++]=rem+64;
        testnum/=26;
    }
    str[ind]='\0';
    start=0;
    end=ind-1;
    while(start<end)
    {
        temp = str[start];
        str[start]=str[end];
        str[end]=temp;
        start++;
        end--;
    }
    //strrev(str);
    printf("%s",str);
   // }
    return 0;
}
```

Q36

Test Case

Input	Output
XVII	17

Weightage - 5

Input	Output
MMMDCCCXCVIII	3898

Weightage - 10

Weightage - 10

Input

```
MMMCMXXXIX
```

Output

```
3939
```

Weightage - 10

Input

```
MMMDCCCLXXXVIII
```

Output

```
3888
```

Weightage - 10

Input

```
MMMMCDXCVIII
```

Output

```
4498
```

Weightage - 10

Input

```
MMCXCV
```

Output

```
2195
```

Weightage - 10

Input

```
MCCCXCVIII
```

Output

```
1398
```

Weightage - 10

Input

```
CMXLVIII
```

Output

```
948
```

Weightage - 10

Input

```
CMXV
```

Output

```
915
```

Weightage - 5

Input

```
MMMMDCCCXXVIII
```

Output

```
4828
```

Weightage - 10

Input

```
MMMDCXLIX
```

Output

```
3649
```

Weightage - 10

Sample Input

```
XLV
```

Sample Output

```
45
```

Solution

```
#include<stdio.h>
//#include<conio.h>

int getValue( char ch)
{
   if( ch == 'I' )
     return 1;
   if( ch == 'V' )
     return 5;
   if( ch == 'X' )
     return 10;
  if( ch == 'L' )
     return 50;
 if( ch == 'C' )
     return 100;
 if( ch == 'D' )
     return 500;
 if( ch == 'M' )
     return 1000;



}
int main()
{
   char str[100]="";
   int num=0,ctr,v1,v2,count=0;
   scanf("%s",str);
   for( ctr = 0 ; str[ctr] != '\0'; )
   {
       v1 = getValue(str[ctr]);

       if( str[ctr+1] != '\0')
       {
       v2 = getValue(str[ctr+1]);

       if( v1 < v2)
       {
     num += (v2-v1);
     ctr+=2;
       }else
       {
     num += v1;
     ctr++;
       }
       }
       else {
       num += v1; ctr++;
       }
     }

   printf("%d",num);
 }
```

```
#include<stdio.h>
//#include<conio.h>

int getValue( char ch)
{
   if( ch == 'I' )
     return 1;
   if( ch == 'V' )
     return 5;
   if( ch == 'X' )
     return 10;
  if( ch == 'L' )
     return 50;
 if( ch == 'C' )
     return 100;
 if( ch == 'D' )
     return 500;
 if( ch == 'M' )
     return 1000;



}
int main()
{
   char str[100]="";
   int num=0,ctr,v1,v2,count=0;
   scanf("%s",str);
   for( ctr = 0 ; str[ctr] != '\0'; )
   {
       v1 = getValue(str[ctr]);

       if( str[ctr+1] != '\0')
       {
       v2 = getValue(str[ctr+1]);

       if( v1 < v2)
       {
     num += (v2-v1);
     ctr+=2;
       }else
       {
     num += v1;
     ctr++;
       }
       }
       else {
       num += v1; ctr++;
       }
     }

   printf("%d",num);
 }
```

Q37

Test Case

Input

```
1000
```

Output

```
444434334
```

Weightage - 10

Input

```
567
```

Output

```
333444333
```

Weightage - 10

Input

Output

33

33343

Weightage - 10

Input

Output

3456

43443333334

Weightage - 10

Input

Output

46474

344343443334344

Weightage - 10

Input

Output

300

33434434

Weightage - 20

Input

Output

482874

443434444333444344

Weightage - 10

Input

Output

648438

3344443343344443444

Weightage - 10

Input

Output

342

34343444

Weightage - 10

Sample Input

Sample Output

10

344

Sample Input

Sample Output

6743

434334344333

Solution

```
#include<stdio.h>
int main()
{
long long int n , start , end , count , ctr;
scanf("%lld" , &n);
long long int arr[n];
start = 0 ;
end = 1;
count = 2;
arr[0] = 3;
arr[1] = 4;
while(count <= n )
{
for(ctr = start ; ctr <= end && count <= n  ; ctr++)
{
    arr[count++] = arr[ctr] * 10LL + 3;
    arr[count++] = arr[ctr] * 10LL + 4;
}
 start = end + 1;
 end = count - 1;
}

printf("%lld" , arr[n-1]);
```

```
#include<stdio.h>
int main()
{
long long int n , start , end , count , ctr;
scanf("%lld" , &n);
long long int arr[n];
start = 0 ;
end = 1;
count = 2;
arr[0] = 3;
arr[1] = 4;
while(count <= n )
{
for(ctr = start ; ctr <= end && count <= n  ; ctr++)
{
    arr[count++] = arr[ctr] * 10LL + 3;
    arr[count++] = arr[ctr] * 10LL + 4;
}
 start = end + 1;
 end = count - 1;
}

printf("%lld" , arr[n-1]);
```

```
return 0;
}
```

```
return 0;
}
```

Q38

Test Case

Input

```
45236  123456  7
```

Output

```
202025
```

Weightage - 10

Input

```
122121  12012  3
```

Output

```
211210
```

Weightage - 10

Input

```
12345  64754  10
```

Output

```
77099
```

Weightage - 10

Input

```
3442   344210  5
```

Output

```
403202
```

Weightage - 10

Input

```
2323  2323  4
```

Output

```
11312
```

Weightage - 10

Input **Output**

63434 32674 8	116330

Weightage - 10

Input **Output**

10110 111 2	11101

Weightage - 10

Input **Output**

56745 67 10	56812

Weightage - 10

Input **Output**

6654 328 9	7083

Weightage - 20

Sample Input **Sample Output**

1010 11001 2	100011

Sample Input **Sample Output**

123 13 4	202

Solution

```
#include<stdio.h>
int main()
{
    int num1 , num2 , base , carry , power, newnum , sum;
    scanf("%d%d%d", &num1 , &num2 , &base);

    power = 1;
    carry = newnum = 0;
    while( num1/power || num2/power )
    {
        sum = (((num1/power)%10) + ((num2/power)%10)) + carry;
        newnum = (sum%base) * power + newnum;
        carry = sum / base;
        power *= 10;
    }
    if(carry)
         newnum = carry * power + newnum;
printf("%d" , newnum);

    return 0;
}
```

```
#include<stdio.h>
int main()
{
    int num1 , num2 , base , carry , power, newnum , sum;
    scanf("%d%d%d", &num1 , &num2 , &base);

    power = 1;
    carry = newnum = 0;
    while( num1/power || num2/power )
    {
        sum = (((num1/power)%10) + ((num2/power)%10)) + carry;
        newnum = (sum%base) * power + newnum;
        carry = sum / base;
        power *= 10;
    }
    if(carry)
         newnum = carry * power + newnum;
printf("%d" , newnum);

    return 0;
}
```

Q39

Test Case

Input **Output**

27	AA

Weightage - 5

Input

```
987654321
```

Output

```
CECGIBQ
```

Weightage - 10

Input

```
556
```

Output

```
UJ
```

Weightage - 5

Input

```
123456789
```

Output

```
JJDDJA
```

Weightage - 10

Input

```
16031994
```

Output

```
AIBCYD
```

Weightage - 10

Input

```
24031995
```

Output

```
AZOHGM
```

Weightage - 10

Input

```
6101965
```

Output

```
MIDNY
```

Weightage - 10

Input

```
1000
```

Output

```
ALL
```

Weightage - 10

Input

```
676
```

Output

```
YZ
```

Weightage - 10

Input

```
16384
```

Output

```
XFD
```

Weightage - 10

Input

```
27122005
```

Output

```
BGICHA
```

Weightage - 10

Sample Input

```
26
```

Sample Output

```
Z
```

Solution

```
#include <stdio.h>
#include<string.h>
int main() {
    //code
    unsigned int testnum=101027545,rem,ind,cases,tc,start,end;
  char str[100],temp;
//  scanf("%u",&cases);
 // for(tc =0 ; tc < cases ; tc++)
 // {
      scanf("%u",&testnum);
  ind=0;
  while(testnum)
  {
      rem = (testnum% 26);
      if(rem==0) {rem=26; testnum--;}
      //printf("%c",rem+64);
      str[ind++]=rem+64;
      testnum/=26;
  }
  str[ind]='\0';
  start=0;
  end=ind-1;
  while(start<end)
  {
      temp = str[start];
      str[start]=str[end];
      str[end]=temp;
      start++;
      end--;
  }
  //strrev(str);
  printf("%s",str);
 // }
    return 0;
}
```

Q40

Test Case

Input

```
1000 10
```

Output

```
True
```

Weightage - 25

Input

```
5000 4
```

Output

```
False
```

Weightage - 25

Input

```
125  3
```

Output

```
False
```

Weightage - 25

Input

```
23546556  4
```

Output

```
False
```

Weightage - 25

Sample Input

```
625  5
```

Sample Output

```
True
```

Sample Input

```
128  5
```

Sample Output

```
False
```

Solution

```
#include<stdio.h>
int isPower(long long int y, long long int x)
{
    // The only power of 1 is 1 itself
    if (x == 1)
        return (y == 1);

    // Repeatedly comput power of x
    long int pow = 1;
    while (pow < y)
        pow *= x;

    // Check if power of x becomes y
    return (pow == y);
}

/* Driver program to test above function */
int main()
{
    long long x,y;
    scanf("%lld %lld",&x,&y);
    if( isPower(x,y) == 1) printf("True");
    else printf("False");
    return 0;
}
```

Section 2 - ZOHO 1

Q1

1234

Solution

Q2

0 1 2 3 4 4 3 2 1 0

Solution

Q3

0

Solution

Q4

4

Solution

Q5 9000

Solution

Q6 96 7

Solution

Q7 8

Solution

Q8 x=2

Solution

Q9 harmr’aharmr’aewlett-packard

Solution

Q10 x = 1 y = 0

Solution

Q11 200

Solution

Q12 a and b are not equal

Solution

Q13 4 3 2 1 0

Solution

Q14 i=-5 j=10

Solution

Q15 Pilots are on strike

Solution

Q16 Stoned

Solution

Q17 tce-cse-b

Solution

Q18 2 3 4 5 6

Solution

Q19 2 1 3 1 4 1 5 1 6 1 7 0

Solution

Q20 x = 3 y = 9 z =27

Solution

Q21 x = 4 z = 1

Solution

Q22 x = 2 y = 1 z = 1

Solution

Q23 harmr’aharmr’aewlett-packard

Solution

Q24 error

Solution

Q25 continue statement not with in a loop

Solution

Q26 cbyebye

Solution

Q27 acerdell

Solution

Q28 -1 -2 -2 -2 -2 1

Solution

Q29 011000000000000000

Solution

Q30 error

Solution

Q31 unpredictable string

Solution

Q32 k = 38

Solution

Q33 81

Solution

Q34 5

Solution

Q35 i=99 a = 0.000000 a=0.000000 i=0 a=3.140000 i=99 i=99 a=3.140000

Solution

Q36 ooho

Solution

Q37 55555

Solution

Q38 k = 36 z = addressofk y = addressofk

Solution

Q39 -1

Solution

Q40 02

Solution

Q41 100 100 100 100

Solution

Q42 43 3

Solution

Q43 a b c d e

Solution

Q44 q before call 2293528 Q after call 2293532

Solution

Q45 recursive infinite function call

Solution

Q46 364

Solution

Q47 recursive call infinite times

Solution

Q48 infinite loop

Solution

Q49 print ascii value followed by corresponding characters

Solution

Q50 2

Solution

Q51 condition not statisfied A=0.700000 b=0.700000

Solution

Q52 runtime error

Solution

Q53 icecream

Solution

Q54 Main's I and j are 10 20 Change's I and j are 100 200 Main's I and j are 10 20

Solution

Q55 x=4.000000 y=2 X=2.200000 y=4

Solution

Q56 x=2.500000 x=15.000000 y=3

Solution

Q57 4

Solution

Q58 x=2.500000 x=15.000000 y=3

Solution

Q59 4 , 4

Solution

Q60 c=5 is getting printed infinite times

Solution

Q61 j=1

Solution

Q62 num=2 n=2 no=2

Solution

Q63 ch = z Ch1 = 122

Solution

Q64 1101791232 21.500000 100 d 0.000000

Solution

Q65 a=25 b=25

Solution

Q66 origin is (0 0) origin is (0 0)

Solution

Q67 i=-1 i=1

Solution

Q68 12

Solution

Q69 i=100 I=101

Solution

Q70 p1

Solution

Q71 TRUE

Solution

Q72 81

Solution

Q73 a= 11 b =3

Solution

Q74 1==1 is TRUE

Solution

Q75 Alphabet

Solution

Q76 1.200000 12.000000 2.400000 24.00000 3.500000 35.000000

Solution

Q77 10 20 30 40 50

Solution

Q78 loop runs 4 times

Solution

Q79 i=-6 j=10 a[1]=11 i=64 m=1 i=2 n=11

Solution

Q80 5 Rahul

Solution

Q81 lvalue required error

Solution

Q82 72 58 114 114

Solution

Q83 Bytes occupied by ch=1 Bytes occupied by a=4 Bytes occupied by b=4

Solution

Q84 4 4 8

Solution

Q85 -2147483648 35 #

Solution

Q86 10 10 10 10 10

Solution

Q87 -647

Solution

Q88 -1

Solution

Q89 2.000000 0.000000

Solution

Q90 0 1 2 3 4

Solution

Q91 0 2

Solution

Q92 0 2 4

Solution

Q93 p = 1 q=1

Solution

Q94 addressofa 0 addressofp addressofa 0 addressofp addressofa 0

Solution

Q95 0 0 0 1 1 1 2 2 2 2 3 3

Solution

Q96 startingaddress 2rowstartingaddress 11

Solution

Q97 0

Solution

Q98 CCCCCCCCCCCCCCCCCCC

Solution

Q99 CCCC ssss mmmm aaaa rrrr tttt

Solution

Q100 n Grunts and Guffaws Dinks Grunts and Guffaws t

Solution

Q101 d = 3 d = 5

Solution

Q102 Malayala

Solution

Q103 4199264

Solution

Q104 Shall we tell the Deputy Director? we tell the Deputy Director? tell the Deputy Director?

Solution

Q105 0 0 0 4

Solution

Q106 6.500000

Solution

Q107 error

Solution

Q108 sunil 30

Solution

Q109 Hacker cr

Solution

Q110 rain =0

Solution

Q111 N B Nagpur Bombay

Solution

Q112 rightest

Solution

Q113 rightest

Solution

Q114 8 4

Solution

Q115 k=-35 k=35

Solution

Q116 1 demo.exe

Solution

Q117 256 0 1

Solution

Q118 2 0 0

Solution

Q119 TTT rrr iii ppp lll eee ttt

Solution

Q120 Salary 1500 Salary1500

Solution

Q121 3 2 515

Solution

Q122 I   am   a   boy

Solution

Q123 a=3.140000 a=   3.14 a= a=   3.1 a=   3

Solution

Q124 Hello Hi Hillo Hello Byelo

Solution

Q125 short leg

                        long leg

                deep fine leg

      backward short leg   legs are the same

Solution

Section 3 - ZOHO

Q1 **Test Case**

Input	Output
1000	444434334

Weightage - 10

Input	Output
567	333444333

Weightage - 10

Input	Output
33	33343

Weightage - 10

Input	Output
3456	43443333334

Weightage - 10

Input	Output
46474	344343443334344

Weightage - 10

Input	Output
300	33434434

Weightage - 20

Input	Output
482874	443434444333444344

Weightage - 10

Input

```
648438
```

Output

```
3344443343344443444
```

Weightage - 10

Input

```
342
```

Output

```
34343444
```

Weightage - 10

Sample Input

```
10
```

Sample Output

```
344
```

Sample Input

```
6743
```

Sample Output

```
434334344333
```

Solution

```
#include<stdio.h>
int main()
{
long long int n , start , end , count , ctr;
scanf("%lld" , &n);
long long int arr[n];
start = 0 ;
end = 1;
count = 2;
arr[0] = 3;
arr[1] = 4;
while(count <= n )
{
for(ctr = start ; ctr <= end && count <= n  ; ctr++)
{
    arr[count++] = arr[ctr] * 10LL + 3;
    arr[count++] = arr[ctr] * 10LL + 4;
}
 start = end + 1;
 end = count - 1;
}

printf("%lld" , arr[n-1]);



return 0;
}
```

Q2

Test Case

Input

```
one two three four five
ree
```

Output

```
one two five four three
```

Weightage - 10

Input

Output

idjbc jdsnc ookjdncv iokndfvkjn iokndcvkln ionfvjnjknv jkndfvjk dncv	idjbc jdsnc kjnvkjddncv vjnfvjkndfkv vkjnfvjknfk jknfvjkfnvkjf

Weightage - 10

Input **Output**

jdgchjdgsjvhgdvjhdfvj hjcvhjdvjhcvdhcvjvc dhvchjdvchvdchjvdjch h cvhjdvchjd	jdgchjdgsjvhgdvjhdfvj hjcvhjdvjhcvdhcvjvc dhvchjdvchvdchjvdjch h

Weightage - 10

Input **Output**

hai hello how are you hai hello ello	hai hello hai you are how hello

Weightage - 10

Input **Output**

wonder wonderlaaa wonderfull wonderful	wonder wonderlaaa wonderfull

Weightage - 10

Input **Output**

wonder wonderlaaa wonderfull hjdsjcfhsdjhcvdcvhjdvchjdvsjc hjdbc wonderful	wonder wonderlaaa hvchjdvcj hjdvchjvdc hdvchjdvjch djhdvc hdbcj

Weightage - 10

Input **Output**

one one one one two three four ne	four three two one one one one

Weightage - 10

Input **Output**

djhvdjch hdbhjv dhjdvc dhcvjhdvcjdhvcjhvdj hjdbvjhdvjdv hjdbhjdv dvc	djhvdjch hdbhjv hjdsjvc hjdchjdvjchv dhvchjdvchjdvc jhdvjhdvhjdv

Weightage - 10

Input **Output**

i love india ve	i india love

Weightage - 10

Input **Output**

hai haiii haiiii aiii	hai haiiii haiii

Weightage - 10

Sample Input **Sample Output**

this is a test sentence st	this is a sentence test

Solution

```
#include<stdio.h>
char * strReverse(char *str)
{
    int start , end ;
    char temp;
    for(end = 0 ; str[end] ; end++);
    for(start = 0 , end-- ; start < end ; start++ , end--)
    {
        temp = str[start];
        str[start] = str[end];
        str[end] = temp;
    }
return str;
}
int substring(char *s1 , char *s2)
{

    int ind1 , ind2 , start = 0 , i , j;
    for(ind1 = 0 ; s1[ind1] ; ind1++)
    {
        if(s1[ind1] == 32)
             start = ind1 + 1;
        else if(s1[ind1] == s2[0])
        {
            for(i= ind1+1 , j = 1 ; s1[i] && s2[j] && s1[i] == s2[j] ; i++ , j++);
            if(s2[j] == 0)
                return start;
        }
    }
    return -1;
}
int main()
{
    char str1[1000] , str2 [1000];
    char *start;
    int ind, st;
    scanf("%[^\n]s" , str1);
    scanf("%s" , str2);
    st = substring(str1 , str2);
    start =  str1 + st ;
    strReverse(start);

    for(ind = st ; str1[ind] ; ind++)
    {
        if(str1[ind] == 32)
        {
            str1[ind] = 0;
            strReverse(start);
            start = str1 + ind + 1;
            str1[ind] = 32;
        }
    }
    strReverse(start);
    printf("%s" , str1);
}
```

Q3

Test Case

Input	Output
1234	1234 1243 1324 1342 1432 1423 2134 2143 2314 2341 2431 2413 3

Weightage - 10

Input	Output
haihel	haihel haihle haiehl haielh haileh hailhe hahiel hahile hak

Weightage - 15

Input

```
wonder
```

Output

```
wonder wondre wonedr wonerd wonred wonrde wodner wodnre wodenr
```

Weightage - 10

Input

```
12345
```

Output

```
12345 12354 12435 12453 12543 12534 13245 13254 13425 13452 13
```

Weightage - 15

Input

```
657547
```

Output

```
657547 657574 657457 657475 657745 657754 655747 655774 655477
```

Weightage - 10

Input

```
123
```

Output

```
123 132 213 231 321 312
```

Weightage - 10

Input

```
gfhg
```

Output

```
gfhg gfgh ghfg ghgf gghf ggfh fghg fggh fhgg fhgg fghg fggh h
```

Weightage - 10

Input

```
!#$%
```

Output

```
!#$% !#%$ !$#% !$%# !%$# !%#$ #!$% #!%$ #$!% #$%! #%$! #%!$ $
```

Weightage - 10

Input

```
<>^V
```

Output

```
<>^V <>V^ <^>V <^V> <V^> <V>^ ><^V ><V^ >^<V >^V< >V^< >V<^ ^
```

Weightage - 10

Sample Input

```
ABC
```

Sample Output

```
ABC ACB BAC BCA CBA CAB
```

Sample Input

```
1234
```

Sample Output

```
1234 1243 1324 1342 1432 1423 2134 2143 2314 2341 2431 2413 3
```

Solution

```c
#include<stdio.h>
void swap(char *x, char *y)
{
    char temp;
    temp = *x;
    *x = *y;
    *y = temp;
}
void permutation(char *str, int l, int r)
{
   int i;
   if (l == r)
     printf("%s ", str);
   else
   {
       for (i = l; i <= r; i++)
       {
          swap((str+l), (str+i));
          permutation(str, l+1, r);
          swap((str+l), (str+i)); //backtrack
       }
   }
}
int main()
{
    char str[100];
    int len;
    scanf("%s" , str);
    for(len = 0 ; str[len] ; len++);
    permutation(str, 0, len-1);
    return 0;
}
```

Q4

Test Case

Input	Output
1000	open = 31 close = 969

Weightage - 10

Input	Output
6456	open = 80 close = 6376

Weightage - 10

Input	Output
234	open = 15 close = 219

Weightage - 10

Input	Output
6576	open = 81 close = 6495

Weightage - 10

Input	Output
775757	open = 880 close = 774877

Weightage - 10

Input

```
5765476
```

Output

```
open = 2401
close = 5763075
```

Weightage - 10

Input

```
1038757
```

Output

```
open = 1019
close = 1037738
```

Weightage - 10

Input

```
88
```

Output

```
open = 9
close = 79
```

Weightage - 10

Input

```
7565
```

Output

```
open = 86
close = 7479
```

Weightage - 10

Input

```
646474
```

Output

```
open = 804
close = 645670
```

Weightage - 10

Sample Input

```
100
```

Sample Output

```
open = 10
close = 90
```

Solution

```
#include<stdio.h>
int main()
{
long long int N , square , ind = 0 , count = 0;
scanf("%lld" , &N);
square = 1;
ind = 2;
while(square <= N)
{
    count++;
    square = ind * ind;
    ind++;
}
printf("open = %lld\nclose = %lld" , count , N-count);
return 0;
}
```

Q5

Test Case

Input

Output

```
3 4
1 2 3 4
5 6 7 8
1 2 3 0
```

```
1 2 3 7 3 2 3
```

Weightage - 10

Input **Output**

```
5 6
1 2 3 4 5 6
7 8 9 0 1 2
1 2 3 4 5 6
```

```
Invalid Path
```

Weightage - 10

Input **Output**

```
5 6
1 2 3 4 5 6
7 8 9 0 1 2
1 2 3 4 5 6
```

```
3 2 1 2 3 9 0 4
```

Weightage - 20

Input **Output**

```
1 7
1 2 3 4 5 6 7
1 1
`````
```

```
1 2 3 4 5 6
```

Weightage - 10

Input **Output**

```
5 5
1 2 3 4 5
6 7 8 9 0
6 7 8 9 0
```

```
4 3 2 7 7 8 7 8 7
```

Weightage - 10

Input **Output**

```
5 5
1 2 3 4 5
6 7 8 9 0
6 7 8 9 0
```

```
Invalid Path
```

Weightage - 10

Input **Output**

```
10 10
1 2 3 4 5 6 7 8  9 0
0 9 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8  9 0
```

```
8 7 6 5 4 5 6 5 6 5 6 5 4 7 4 5 4 5 4 7 4 7 8 7 4 5 6 5
```

Weightage - 10

Input **Output**

```
10 10
1 2 3 4 5 6 7 8  9 0
0 9 8 7 6 5 4 3 2 1
```

```
Invalid Path
```

Weightage - 10

Input **Output**

```
3 3
1 2 3
3 2 1
1 2 3
```

```
1 2 3 1 2 3 1 2 3 1 3 2 3 2 3 2 3 2 3 2 3 2 3
```

Weightage - 10

Sample Input **Sample Output**
```

5 5 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5	8 3 4 5 0 9 8 3 4 5 0 5

Solution

```
#include<stdio.h>
#include<malloc.h>
#define isBoundC(col) (col >= 0 && col < M)
#define isBoundR(row)  (row >= 0 && row < N)
int main()
{
int *path , count = 0;
int N , len , M , row , col , flag,startrow , startcol , ind;
char str[100];
scanf("%d%d" ,&N,&M);
int arr[N][M];
for(row = 0 ; row < N ; row++)
{
    for(col = 0 ; col < M ; col++)
        scanf("%d" , &arr[row][col]);
}
scanf("%d%d",&startrow ,&startcol);
scanf("%s", str);
for(len = 0 ; str[len] ; len++);
path = (int*)calloc(len+1 , sizeof(int));
if(isBoundR(startrow-1) && isBoundC(startcol-1))
{
    startrow--;
    startcol--;

    path[count++] =  arr[startrow][startcol];

    for(ind = 0 , flag = 0 ; flag !=1 && str[ind] ; ind++)
    {
        switch(str[ind])
        {
            case '>' : if(isBoundC(startcol+1))
                            path[count++] = arr[startrow][++startcol];
                       else flag = 1 ;
                       break;
            case '<' : if(isBoundC(startcol-1))
                            path[count++] = arr[startrow][--startcol];
                       else flag = 1 ;
                       break;
            case '^' : if(isBoundR(startrow-1))
                            path[count++] = arr[--startrow][startcol];
                       else flag = 1 ;
                       break;
            case 'v' : if(isBoundR(startrow+1))
                            path[count++] = arr[++startrow][startcol];
                       else flag = 1 ;
                       break;
        }
    }

if(flag == 1 || count == 0)
    printf("Invalid Path");
else
{
    for(ind = 0 ; ind < count ; ind++)
      printf("%d ", path[ind]);
}
}

return 0;
}
```

Q6

Test Case

Input	Output
6 abc cab abcd bac dcba hdjd	cab bac abc dcba abcd hdjd

Weightage - 10

Input

```
10
abc  cab  dhfh  fhfhf  abcd  dcba  dcbaa  aaa  cadb  bacd
```

Output

```
cab  abc
dcba  cadb  bacd  abcd
dhfh
fhfhf
```

Weightage - 10

Input

```
7
hjfdjf  hfdjhfd  jhfjfg  hjfdf  hjdfjdf  hfd   hjfd
```

Output

```
hjfdjf
hfdjhfd
jhfjfg
hjfdf
```

Weightage - 10

Input

```
5
hai  aih  iah  hello  elloh
```

Output

```
aih  iah  hai
elloh  hello
```

Weightage - 10

Input

```
4
abc  fhf  jdfjg  jd
```

Output

```
abc
fhf
jdfjg
jd
```

Weightage - 10

Input

```
6
abcdef  fedcab  abcd  dcba  abc  cab
```

Output

```
fedcab  abcdef
dcba  abcd
cab  abc
```

Weightage - 10

Input

```
3
nwlrbbmqbhcdarzowkkyhiddqscdxrjmowfrxsjybldbefsarc  qscdxrjmowfrxs
```

Output

```
qscdxrjmowfrxsjybldbefsarcnwlrbbmqbhcdarzowkkyhidd  nwlrbbmqbhcdar
hjfjdfhj
```

Weightage - 10

Input

```
5
nwlrbbmqbhcdarzowkkyhiddqscdxrjmowfrxsjybldbefsarcbynecdyggxxpklo
```

Output

```
wfrxsjybldbefsarcbynecdyggxxpklorellnmpapqfwkhonwlrbbmqbhcdarzowl
ddqscdxrjmowfrxsjybldbefsarcnwlrbbmqbhcdarzowkkyhi  nwlrbbmqbhcda
jdfjhfjkg
```

Weightage - 10

Input

```
7
hai  hello  hai  hello  hai  hello  fhf
```

Output

```
hai  hai  hai
hello  hello  hello
fhf
```

Weightage - 10

Input

```
5
jfjf  jjfjg  jgg  abc  cba
```

Output

```
cba  abc
jfjf
jjfjg
jgg
```

Weightage - 10

Sample Input

```
5
tar rat banana atr nanaba
```

Sample Output

```
rat atr tar
nanaba banana
```

Solution

```
#include<stdio.h>
#include<malloc.h>
int strLen(char *str)
{
    int ind;
    for(ind = 0 ; str[ind ] ; ind++);
return ind;
}
int main()
{
char str[100][100];
int N , i , j , set = 0 , ind , ind1 , len , len1 , sum;
int *there;
scanf("%d" , &N);
int result[N] , count = 0;
for(ind = 0 ; ind < N ; ind++)
    scanf("%s" , str[ind]);

for(ind = 0 ; ind < N ; ind++ , set = 0)
{
    if(str[ind])
         len = strLen(str[ind]);
    else
        continue;
    for(ind1 = ind + 1 ; ind1 < N ; ind1++)
    {
        if(str[ind1])
            len1 = strLen(str[ind1]);
        else
            continue;
        if(len == len1)
        {
             there = (int*)calloc(26 , sizeof(int));
             for(i = 0 ; i < len ; i++)
                 there [ str[ind][i] - 97 ]++;
             for(j = 0 ; j < len ; j++)
             {
                 if(there[str[ind1][j] - 97])
                     there[str[ind1][j] - 97]--;
                 else
                    break;
             }
             for(i = 0 , sum = 0; i<26 ; sum+= there[i++]);
             if(sum == 0 && str[ind1][0])
             {
                printf("%s " , str[ind1]);
                str[ind1][0] = 0;
                set =1;
             }

        }

    }
    if(set ==1 && str[ind][0])
    {
         printf("%s", str[ind]);
         printf("\n");
    }

    else if(str[ind][0])
            result[count++] = ind;
}
for(i = 0 ; i < count ; i++)
    printf("%s\n" , str[result[i]]);
return 0;
}
```