CHITKARA UNIVERSITY

2023-2024

Mock Questions For Sessional Test III –March, 2023

Roll No: …… [Total No. of Pages:]

Programme: B.E. (CSE) **Time: 60 minutes**

Course Title: Problem Solving using C

Course Code: CS153 **Max. Marks: 4**

General Instructions:

- Follow the instructions given in each section.

Section - A

(Each question carries 1 mark)

Q1: What is the maximum number of members that can be defined in a structure in C?

a)512 b)256

c)1024 **d)There is no maximum limit**

Q2: Which of the following statements is true about accessing structure members?

a)Structure members are accessed using the '.' operator.

b) Structure members are accessed using the '&' operator.

c)Structure members are accessed using the '->' operator.

d) Structure members are accessed using the '*' operator.

Q3: Structure cannot have _____ as members.

a)Functions b)Structures

c)Both a & b d) none of the above

Q4: What will be the output of the following code?

```
#include <stdio.h>
int main() {
int arr[4] = {10, 20, 30, 40};
int sum = 0;
for(int i = 0; i < 4; i++) {
   sum += arr[i];
```

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

a) 10 b) 70

c) 100 d) 20

Q5: How can you access the second element of an array using pointer notation?

a) *(ptr+1) b) struct {int no;}

c) struct mystruct{ int no;}; d) struct mystruct{ int no;}

Q6: The string in C are represented in actual memory as _____ ?

a) Linked List of Characters b) An object of String Class

c) An object of String structure **d) An array of characters**

Q7: Which of the following is a correct way to declare macros in C ?

a) #define int

b) #define int i

c) #define i

d) #define i int

Q8: What is the size of the int data type in C (in bytes) ?

a) 1

b) 2

c) 4

d) 8

Q9: What will be passed if you pass a variable to a function?

a) address of variable

b) Copy of variable

c) Reference of variable

d) none of the above

Q10: ____ is used to assign names to integral constants?

a) Union **b) Enum**

c) Structure d) Both a and b

Q11: Choose the odd one out from the following.

a) Typedef

b) Structure

c) Integer

d) Union

Q12: What is the use of putw() ?

a) Reads an integer from file

b) Writes an integer to file

c) Writes a character into the file

d) Reads a character from file

Q13: How can Ram open a text file in read and write modes in c?

a) fopen("demo.txt","a+");

b) fopen("demo.txt", "ab");

c) fopen("demo.txt", "wb+");

d) fopen("demo.txt", "ab+");

Q14: Is the given structure a valid structure in C?

```
struct student
{
        char name[50]="ram kumar";
        int roll = 12;
        char grade ='B';
        int marks[3] ={ 78,89,99};
        float avg = 99.78;
}stu;
```

a) No

b) Yes

c) Structure members cannot be initialized with declaration in C.

d) A and C

Q15: The total memory size occupied by the below union is?

```
        union student
        {
                int roll;
                char grade;
                float avg;
                int arr[5];
        }stu;
```

a) 19 bytes

b) 29 bytes

c) 13 bytes

d) 20 bytes

Q16: Which is the false statement about typedef in c?

a) This keyword is used to redefine the name of an already existing variable.

b) You cannot use typedef to give a name to your user defined data types.

c) It can be used with arrays to declare any number of variables.

d) B and C

Q17: Which is the true fact about enum in c?

a) Two enum names can have same value.

b) We can assign values to some name in any order.

c) The values assigned to the enum names must be integral constants and character constants.

d) A and B

Q18: Recursion has the following properties :

a) Used with loops only.

b) Terminates when the base case becomes false.

c) Every recursive call needs extra space in the stack memory.

d) Every iteration does not require any extra space.

Q19: Which of the following return-type is valid for a structure in C?

a) int

b) void

c) char

d) None of the above.

Q20: What will be the output of the given code?

```
enum friends { ram, ravi, Raja=12, ramesh };

int main()

{

        enum friends name;

        name = ramesh;

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

        return 0;

}
```

a) 3

b) 13

c) 4

d) None of the above

Q21: What is the default value of the first member of an enum?

a) 0 b) 1

c) -1 d) undefined

Q22 Which of the following is a requirement for a recursive function in C?

a) The function must have a base case

b) The function must be declared as static

c) The function must have a void return type

d) The function must take no arguments

Q23: What is tail recursion in C?

a) A recursive function where the recursive call is the first statement in the function

b) A recursive function where the recursive call is the last statement in the function

c) A recursive function where the recursive call is not allowed

d) A recursive function where the base case is not defined

Q24: What is the output of the following code snippet?

```
#include <stdio.h>
typedef int example;
int main() {
    example x = 10;
    printf("%d\n", x);
    return 0;
}
```

a) 10

b) example

c) An error will occur

d) Nothing will be printed

Q25: What is the operator used to access a member of a structure through a pointer?

a) &

b) *

c) ->

d) .

Q26: What is the syntax for declaring a structure pointer variable in C?

a) Person p;

b) struct Person *p;

c) Person *p;

d) *p Person;

Q27: What is the keyword used to declare a C file pointer.?

a). file

b). FILE

c). FILEFP

d). filefp

Q28: What is the output of the following code snippet?

```
#include <stdio.h>

enum color {
    RED,
    GREEN,
```

```
    BLUE
};
int main() {
    enum color c = GREEN;
    printf("%d", c);
    return 0;
}
```

a) 0

b) 1

c) 2

d) GREEN

Q29 How is the size of a union determined?

a) By the size of the largest member

b) By the size of the smallest member

c) By the sum of the sizes of all members

d) By the average size of all members

Q30: How can you disable structure padding in C?

a) Using the #pragma pack directive

b) Using the #pragma align directive

c) Using the #pragma pad directive

d) There is no way to disable structure padding

Section - B

(Each question carries 2 marks)

Q31 What will be the output of the following code ?

```
#include <stdio.h>
#include<string.h>
void convertToNo() {
 char ch[10] = "abcde";
 int ans = 0;
for(int i = 0; i < strlen(ch); i++) {
  ans += (ch[i] - 'a');
}
printf("%d", ans);
}
int main() {
  convertToNo();
 return 0;
}
```

a) 9 b) 11

c) 10 d) 6

Q32: What will be the output of the following code ?

```
#include<stdio.h>
    enum statusCode                {
    ok=200, notFound=404, serverError=500
};
int main()
{
 statusCode code;
code=ok;
printf("%d",ok);
}
```

a) 200 b) 404

c) 500 d) Syntax Error

Q33: What will be the output of the following code ?

```
 #include<stdio.h>
void func(int n)
{
    if(n==0)
       return;
    else
      {
         char ch='a';
```

```
        printf("%c",ch+n-1);
        func(n-1);
    }
}
int main()
{
    int a;
   func(4);
}
```

a) edcb b) edcba

c) dcba d) none of the above

Q34: What will be the output of the following code ?

```
    #include<stdio.h>
    int func(int n)
    {
      if(n<0)
      return 0;

      return(n+func(n-2));

    }
```

```
int main()
{
 int n=3;
 printf("%d",func(n));
}
```

a) 5 b) 3

c) Undefined Behaviour **d) 4**

Q35: What will be the output of the following code?

```
#include<stdio.h>
int func(int x)
{
    if(x==1)
    return 1;
    int ans=x*func(x-1);
    return ans;
}
int main()
{
     int n;
     n=func(3);
     printf("%d",n);
}
```

a) 9 b) 3

c) 6 d) 24

Q36 Which of the following is **not a correct** file opening mode ?

a) r b) r+

c) rb **d) r-**

Q37 What will be the output of the following code ?

```
#include <stdio.h>
void calculate(int x) {
   printf("%d", (x << 1) + (x >> 1));
}
int main() {
   int no=4;
   calculate(no);
   return 0;
}
```

a) 8 b) 16

c) 10 d) 12

Q38 Which data structure is used to handle recursion i.e. function calls in C?

a) Stack b) Queue

c) Heap d) Linked List

Q39 Which of the following is **not correct** about structures?

a) variables of structure can be created b) pointer of structure can be created

c) function can be created inside structures d) none of the above

Q40 What will be the output of the following code ?

```
#include <stdio.h>
#include<string.h>
void print() {
   char str[] = "Turing Block";
   printf("%s ", str);
   char cp[40];
   strcpy(cp, str);
   printf("%s", cp);
}
int main() {
   print();
   return 0;
}
```

a) Turing Block **b) Turing Block Turing Block**
c) Turing d) Block

Q41: What will be the output of the given code on 64-bit machine?

```
#include<stdio.h>

struct student

{

   int a;

   char b;

} stu;

int main()

{

   printf("%d",sizeof(stu));

   return 0;

}
```

a) 8

b) 6

c) 5

d) 3

Q42: What will be the output of the following code?

```
union student
{
   int roll;
  char grade;
  float avg;
  double val;
};
int main()
{
    stu.roll=12;
    stu.grade='A';
    stu.avg=78.3234;
   printf("%.2f\n",stu.avg);
   return 0;
}
```

a) 78.32

b) 78.3234

c) 78.322

d) Code has an error.

Q43: What will be the output of the below code ?

If user inputs are:

4

2

```
#include <stdio.h>
int fn(int n1, int n2);
int main()
{
    int base, a, result;
    scanf("%d%d", &base,&a);
    result = fn(base, a);
    printf("%d",result);
    return 0;
}
int fn(int base, int a)
{
    if (a != 0)
    return (base * fn(base, a - 1));
    else
    return 1;
```

```
}
```

a) 8

b) 16

c) 32

d) 64

Q44: Predict the output of the given code below.

```
#include<stdio.h>

enum subject { hindi=29, english, maths=29, Computer };

int main()

{

        enum subject name;

        name = english;

        printf("%d",name);

        name = computer;

        printf("%d",name);

        return 0;

}
```

a) 24

b) 2929

c) 13

d) 3030

Q45: What will be the output of the below code ?

```
#include<stdio.h>

struct student

{

    char name[50];

    int roll;

    char grade;

    int marks[3];

    float avg;

};

int main()

{

    struct student stu;

    stu={.marks[0]=88,.marks[1]=87,.marks[2]=90};

    printf("%d",stu.marks[2]);

    printf("%d",stu.marks[1]);

    printf("%d",stu.marks[0]);

    return 0;

}
```

a) 908788

b) 888790

c) 878890

d) It will give an error while compiling.

Q46: What is the output of the following code snippet?

```
#include <stdio.h>

void print_num(int n)
{
   if (n == 0)
      return;
   else
   {
      printf("%d ", n);
      print_num(n - 1);
      printf("%d ", n);
   }
}

int main()
{
   print_num(3);
   return 0;
}
```

a) 3 2 1 1 2 3
b) 1 2 3 3 2 1
c) 3 2 1 3 2 1
d) 1 3 2 1 2 3

Q47: what will be the output of the following code snippt

```
void print_arr(int arr[], int n)
{
   if (n == 0)
      return;
   else
   {
      print_arr(arr, n - 1);
      printf("%d ", arr[n - 1]);
   }
}
int main()
{
    int arr[] = {1, 2, 3, 4, 5};
   print_arr(arr, 5);
}
```

a) 1 2 3 4 5

b) 5 4 3 2 1

c) 2 3 4 5 1

d) 5 1 4 2 3

Q48: What is the output of the following code snippet?

```
#include <stdio.h>
struct Person {
   char name[20];
   int age;
};
```

```
int main() {
    struct Person p1 = { "Alice", 25 };
    struct Person p2 = p1;
    p2.name[0] = 'B';
    printf("%s\n", p2.name);
    return 0;
}
```

a) Alice
b) Bob
c) Blice
d) Runtime error

Q49: What is the size of the following structure on a 32-bit system?

```
struct Example {
    char c;
    double d;
};
```

a) 8 bytes
b) 9 bytes
c) 12 bytes
d) 16 bytes

Q50: What is the output of the following code snippet?

```
#include <stdio.h>

struct Example {
    int x;
    int y;
};

int main() {
```

```
    struct Example ex = {10, 20};
    struct Example *p = &ex;
    printf("%d\n", p->y);
    return 0;
}
```

a) 10
b) 20
c) An error will occur
d) Nothing will be printed

Section – C

(Each question carries 5 marks)

Q51. Vikram is trying to write a combination of opening and closing braces. He observed that some expressions are balanced whereas some are not. A balanced parenthesis expression has the following properties. Open brackets must be closed by the same type of brackets. Open brackets must be closed in the correct order.Every close bracket has a corresponding open bracket of the same type.

Help Vikram to generate all the n pairs of balanced parenthesis.

Input Format:
First line will take an integer n

Output Format:
Print the valid expressions of balanced parenthesis in n lines.

Input :
2
Output :
(())
()()

Explanation : Vikram has to print all the combinations with n pairs of balanced parenthe sis in this case n=2. So he has to print all the combinations with 2 pairs of opening and closing brackets. -

Input:
3
Output :
((()))
(()())
(())()
()(())
()()()

Explanation : Vikram has to print all the combinations with n pairs of balanced parenthe sis in this case n=3. So he has to print all the combinations with 3 pairs of opening and closing brackets. -

Input :
4
Output:
(((())))
((()()))
((())())
((()))()
(()(()))
(()()())
(()())()
(())(())
(())()()
()((()))
()(()())
()(())()

Explanation : Vikram has to print all the combinations with n pairs of balanced parenthe sis in this case n=4. So he has to print all the combinations with 4 pairs of opening and closing brackets i.e. all the balanced parenthesis with 4 opening brackets and 4 closing - brackets.

Solution

```
#include <stdio.h>
int parenthesis(int n,int open,int close,int idx,char* ans)
{

     if(idx==2*n)
     {
        printf("%s\n",ans);
     }

       if(open<n)
       {
          ans[idx]='(';
          parenthesis(n,open+1,close,idx+1,ans);
       }

       if(close<open)
       {
          ans[idx]=')';
          parenthesis(n,open,close+1,idx+1,ans);
       }

}
int main() {
    char ans[100]="";
    int n;
    scanf("%d",&n);
  parenthesis(n,0,0,0,ans);
}
```

Q52. James want to climb n steps from ground level. At each step he can either take a jump of 1 step or a jump of 2 steps or a jump of 3 steps. Find the total number of in which James can reach the nth step.

Input : 2
Output: 2
Explanation : James can reach the 2nd step in 2 ways if at each step he can take a jump of 1,2 or 3. In this reach 2nd step by the following ways step 0 to step 2 directly or 0 to 1 then step 1 to step 2

Input: 3
Output: 4
Explanation : James can reach the 3rd step in 4 ways if at each step he can take a jump of 1,2 or 3.

Input: 4
Output: 7
Explanation : James can reach the 4th step in 7 ways if at each step he can take a jump of 1,2 or 3.

Solution 22:

```
#include<stdio.h>
int ways(int n)
{
   if(n==0)
   return 1;

   if(n<0)
   return 0;

   return ways(n-1)+ways(n-2)+ways(n-3);
}
int main(){
   int n;
   scanf("%d",&n);
   printf("%d",ways(n));
}
```

Q53. Rishi sir has given a fun task to Ravi. The task is to calculate the sum of natural numbers in a given range x, y using a recursive function and print the sum. if the sum is even then print message **HAPPY** otherwise print message **SAD**.

Input Format

Input N = Number of Test Cases, followed by x and y.

x is lower limit and y is upper limit

Output Format

First line will print sum of natural number and next line will print message HAPPY or SAD.

Sample Input

3

1 1000

50 100

100 150

Sample Output

500500

HAPPY

3825

SAD

6375

SAD

Explanation :
The sum of natural numbers from 1 to 1000 is 500500, which is even, hence the message is HAPPY.

The sum of natural numbers from 50 to 100 is 3825, which is odd, hence the message is SAD.

The sum of natural numbers from 100 to 150 is 6375, which is odd, hence the message is SAD.

```
#include <stdio.h>
int fun_sum(int x, int y)
{
if(x == y)
return x;
else
return x + fun_sum(x + 1, y);
}

int main()
{
int n,x,y,sum,i;
scanf("%d",&n);
for(i=1; i<=n; i++)
{
scanf("%d%d",&x,&y);
sum = fun_sum(x, y);
printf("%d\n",sum);
(sum%2==0)?printf("HAPPY\n"):printf("SAD\n");
}
return 0;
}
```

Q54 Take as input S, a string. Write a function that replaces every even character with the character having just higher ASCII code and every odd character with the character having just lower ASCII code. Print the value returned.

Input Format : Accept a String
Output Format: Print a String

Input: abcg
Output: badf
Explanation: Every character at even position is replaced with next character and every character at odd position is replaced with its just previous character.

Input : cdef
Output : dcfe

Input : cbae
Output: dabd
Solution:

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

    char str[1000];

    scanf("%s",str);

    for(int i=0;i<strlen(str);i++)
    {
        if(i%2==0)
        str[i]++;

        else
        str[i]--;

    }

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

Q55 Take as input S, a string. Write a function that does basic string compression. Print the value returned. E.g. for input “aaabbccds” print out a3b2c2d1s1.

Input Format:
Accept a String
Output Format:
Print a String

Input :
aaabbccds
Output :
a3b2c2d1s1

Explanation : In the given sample test case 'a' is repeated 3 times consecutively, 'b' is repeated twice, 'c' is repeated twice and 'd and 's' occurred only once.

Input :
aabbc
Output:
a2b2c1
Explanation : In the given sample test case 'a' is repeated 2 times consecutively, 'b' is repeated twice, 'c' has occurred once.

Input:
ciia
Output:
c1i2a1
Explanation : In the given sample test case ‘c' has occurred once, 'i' is repeated 2 times consecutively, 'a' has occurred once.

Solution :
```
#include<stdio.h>
#include<string.h>
void func(char * str, char* ans)
{
    int k=0;
    int n=strlen(str);

   for(int i=0;i<n;i++)
   {
        int cnt=1;
        char ch=str[i];

      while(str[i]==str[i+1])
      {
         cnt++;
         i++;
       }

      ans[k++]=ch;
      char cntString[100];
      sprintf(cntString,"%d",cnt);
      for(int j=0;j<strlen(cntString);j++)
      ans[k++]=cntString[j];
      //  printf("%c%d",ch,cnt);
   }
   ans[k]='\0';

}
int main(void) {

   char str[1000];
   scanf("%s",str);
   int n=strlen(str);
   char ans[1000];
   func(str,ans);

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

Q56. Deepshikha wants to count a number of 1s in the binary representation of an integer. Write a function named **count1s** that can take an integer value as an argument and return a number of 1s in the binary representation of a given integer value.

Input Format

Input N = Number of Test Cases, followed by N numbers.

Output Format

Number of 1s in each number in a new line.

Sample Input

3

5

4

15

Sample Output

2

1

4

Explanation :

The binary of 5 is 1001 hence number of 1s = 2.
The binary of 4 is 1000 hence number of 1s = 1.

The binary of 15 is 1111 hence number of 1s = 4.

```
#include<stdio.h>

int count1s(int num)
{
    int count=0;

                while(num>0)
                {
                        if(num%2==1)
                        {
                                count++;
                        }
                        num=num/2;
                }

                return count;
}

int main()
{
        int n,i;

        scanf("%d",&n);
        for(i=1; i<=n; i++)
        {
       int num;
            scanf("%d",&num);
                printf("%d\n",count1s(num));
        }
        return 0;
}
```

Q 58 Given a string s consisting of lowercase English letters, return the first letter to appear twice.

Note:

A letter a appears twice before another letter b if the second occurrence of a is before the second occurrence of b.
s will contain at least one letter that appears twice.

Sample Input 1:
abccbaacz
Sample Output 1:
c

Sample Input 2:
abcdd
Sample Output 2:
d

```
#include<stdio.h>

char repeatedCh(char * s) {

	char alpha[26] = {0};

	register unsigned char i = 0;
	while(s[i] != '\0') {
		alpha[s[i] - 'a']++;
		if(alpha[s[i] - 'a'] > 1)
			break;
		i++;
	}
	return s[i];
}

int main()
{
   char s[1000];
   gets(s);
   printf("%c",repeatedCh(s));
}
```

Q 59. Break the number

You are given a 0-indexed integer array nums. An index i is part of a hill in nums if the closest non-equal neighbors of i are smaller than nums[i]. Similarly, an index i is part of a valley in nums if the closest non-equal neighbors of i are larger than nums[i]. Adjacent indices i and j are part of the same hill or valley if nums[i] == nums[j].

Note that for an index to be part of a hill or valley, it must have a non-equal neighbor on both the left and right of the index.

Return the number of hills and valleys in nums.

Input Format
first line will take an input integer n
next n lines will have elements of array

Output Format
the number of hills and valleys in given array

Sample Input 1 :
6
1 2 3 3 2 1

Sample Output 1:
1

Sample Input 2 :
3
1 2 2

Sample Output 2:
0

Ans code :-

```
#include<stdio.h>
int countHillValley(int* nums, int n)
{
   int i = 1, j = 0, count = 0;
   int temp[n];

   while(i < n)
   {
      temp[j++] = nums[i-1];
      while(i < n && nums[i] == nums[i-1]) i++;
```

```
        i++;
    }
    //in case last one is one non repeating element
    if(i <= n)
        temp[j++] = nums[i-1];

    int l = j;
    i = 1;
    while(i < l-1)
    {
        if(temp[i] > temp[i+1] && temp[i] > temp[i-1]) count++;
        if(temp[i] < temp[i+1] && temp[i] < temp[i-1]) count++;
        i++;
    }
    return count;
}

int main()
{
   int n;
   scanf("%d",&n);
   int arr[n];
   for(int i=0;i<n;i++)
   scanf("%d",&arr[i]);

   printf("%d",countHillValley(arr,n));
}
```

Section – D

(Q 25: Question carries 10 marks)

Q60. Lakshay is learning recursion. His instructor Naveen bhaiya has given him a problem to print all the permutations of a given string. As Lakshay is just learning recursion and backtracking, you have help him find all the possible permutations of the given string.

Input Format:
Accept a string from user

Output Format:
In next n lines Print all the permutations of this given string.

Input:
ab
Output:
ab
ba

Input:
abc
Output:
abc
acb
bac
bca
cba

cab

Input:
a
Output:
a

Solution 60:

```
#include <stdio.h>
#include<string.h>
void swap(char* str,int i,int j)
{
   char tmp=str[i];
   str[i]=str[j];
   str[j]=tmp;
}
void permutations(char* str,int idx)
{

   if(idx==strlen(str))
   {
      printf("%s\n",str);
      return;
   }

   for(int i=idx;i<strlen(str);i++)
   {
      swap(str,i,idx);
      permutations(str,idx+1);
       swap(str,i,idx);
   }
}
int main() {
   char str[100];
   scanf("%s",str);
   permutations(str,0);
}
```

Q61. Ishan Raj Sir wants to save the student information such as name, roll no, and marks in three subjects of n students and wants to print out the name and roll no of those students who have got total percentage less than the given percentage criteria.
Your task is to create a structure and members of the structure include name, roll no, marks in three subjects, and percentage.

Note:

1. Assume marks and percentage as an integer value and for the rest of the members you use your wisdom to decide the datatype.

2. The maximum mark for each subject is 100.

3. Use your wisdom to calculate the percentage of marks obtained by students in all three subjects.

Constraints

1<=n<=100.

0<=mark<=100.

1<=roll no <=100.

Input Format:
Order of inputs are as follow :
1. Total student – n
2. Passing percentage -percentage
3. Name of student – name
4. Roll no – rollno.
5. Marks – mark1 mark2 mark3

Repeat the order of inputs line 3 to 5 for n students.

Output Format:
Print name - name
Print roll no. - rollno

Sample Input:
3
40
ram
11
98
33
34
mohan

12
65
77
88
shivam
13
35
40
40

Sample Output:
shivam
13

Explanation :
For student ram, roll no – 11
mark1 = 98, mark2 = 33, mark3 = 34.
Total percentage = (98+33+34)100/300
= 55 % which greater than 40 %.
For student mohan, roll no – 12
mark1 = 65, mark2 = 77, mark3 = 88.
Total percentage = (65+77+88)100/300
= 76 % which is greater than 40 %.
For student shivam, roll no – 13
mark1 = 35, mark2 = 40, mark3 = 40
Total percentage = (98+33+34)100/300
= 38 % which is less than 40 %.
Hence output is shivam 13.

Solution 61:

```
#include<stdio.h>

struct student
{
char name[50];
int marks[3];
int rollno;
int percentage;

} stu[100];

int main()
{
```

```
int n,i,j,passing_per;

scanf("%d",&n);
scanf("%d",&passing_per);

for(i=0; i<n;i++)
{
scanf("%s",stu[i].name);
scanf("%d",&stu[i].rollno);
int sum=0;
for(j=0; j<3; j++)
{
scanf("%d",&stu[i].marks[j]);
sum=sum+stu[i].marks[j];
}
stu[i].percentage = (sum*100)/300;
}

for(i=0; i<n;i++)
{
if(stu[i].percentage<passing_per)
{
printf("%s\n",stu[i].name);
printf("%d\n",stu[i].rollno);
}
}

return 0;
}
```

Q62 Given a string s and an integer k, reverse the first k characters for every 2k charac ters counting from the start of the string.

-

If there are fewer than k characters left, reverse all of them. If there are less than 2k but greater than or equal to k characters, then reverse the first k characters and leave the other as original.

Sample input 1
Abcdefg
2

sample output 1
bacdfeg

Sample input 2
abcd
2
sample output 2
bacd