PPS CT2 - Prev year QP Set-3

Programming For Problem Solving (SRM Institute of Science and Technology)

SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing

DEPARTMENT OF COMPUTING TECHNOLOGIES

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

Academic Year: 2022-23 (ODD)

Test: CLAT- 2
Course Code & Title: 21CSS101J / Programming for Problem Solving – Answer Key
Year & Sem: I / I **Max. Marks: 50**

NOTE:

- **For each 5 mark question, 2 marks logic (as seen in program irrespective of syntax errors) and 3 marks for program.**
- **Based on number of syntax errors in program, program marks (3) may be allotted.**
- **There is no need to write #inlcude<stdio.h> for all programs**
- **There are many ways of writing the same program!**

Part - A
(1 x 25 = 25 Marks)
Instructions: This section has only ONE question with internal choice.

Q. No	Question	Marks	BL	CO	PO	PI Code

1a	**(i) Problem:**
	There is a group of N friends who wish to enrol in a course together. The course has a maximum capacity of M students that can register for it. If there are K other students who have already enrolled in the course, determine if it will still be possible for all the N friends to do so or not.
	Input Format:
	• Each test case consists of a single line containing three integers N, M and K - the size of the friend group, the capacity of the course and the number of students already registered for the course.
	Output Format:
	• For each test case, output Yes if it will be possible for all the N friends to register for the course. Otherwise output No.
	Input: **Output:**
	3
	2 50 27 Yes
	5 40 38 No
	100 100 0 Yes
	Solution:
	`#include <stdio.h>`
	`int main() {`
	`int t;`
	`scanf("%d",&t);`
	`while(t--)`
	`{`
	`int n,m,k;`
	`scanf("%d %d %d",&n, &m, &k);`
	`m-=k;`
	`if(n<=m)`
	`printf("YES\n");`
	`else`
	`printf("NO\n");`
	`}`
		5	1	2	PO1	1.7.1

The maximum rebatable amount for any

	return 0;
	}
	
	(ii) Problem:
	Karan bought car insurance. The policy of the insurance is:
	
	• Karan's car meets an accident and required Rs Y lakhs for repairing. Determine the amount that will be rebated by the insurance company.
	Input Format:
	• The first and only line of each test case contains two space-separated integers X and Y.
	Output Format:
	• For each test case, output the amount (in lakhs) that will be rebated by the insurance company.
	
	Input: **Output:**
	4
	5 3 3
	5 8 5
	4 4 4
	15 12 12
	
	Solution:
	
	#include <stdio.h>
	int main() {
	int n;
	scanf("%d",&n);
	while(n>0)
	{
	int x,y,min;
	scanf("%d %d",&x,&y);
	if (x<y)
	min=x;
	else
	min=y;
	printf("%d\n",min);
	n--;
	}
	return 0;
	}
		5	**2**	**2**	**PO2**	**2.5.2**

	(iii) Problem: Rahul has N friends. He promised that he would gift a pair of shoes (consisting of one left shoe and one right shoe) to each of his N friends. Rahul was about to go to the marketplace to buy shoes, but he suddenly remembers that he already had M left shoes. What is the minimum number of extra shoes that Rahul will have to buy to ensure that he is able to gift a pair of shoes to each of his N friends? **Input Format:** • Each test case contains two integers N and M - the number of Rahuls's friends and the number of left shoes Rahul has. **Output Format:** For each test case, output the minimum number of extra shoes that	**5**	**2**	**2**	PO2	**2.5.2**

	Rahul will have to buy to ensure that he is able to get N pairs of shoes.
	Solution:
	#include<stdio.h>
	int main()
	{
	int n,m,t;
	scanf("%d",&t);
	for(int i=1;i<=t;i++)
	{
	scanf("%d%d",&n,&m);
	if(m<=n)
	{
	int k=(2*n)-m;
	printf("%d",k);
	}
	else
	{
	printf("%d",n);
	}
	printf("\n");
	}
	}
	
	(iv) Problem:
	In a test, there are N problems, each carrying X marks. In each problem, Amala either received X marks or 0 marks. Determine whether is it possible for her to achieve exactly Y marks.
	Input Format:
	• Each test case consists of three integers N, X and Y, the number of problems, the maximum score for each problem, and the score Amala wants.
	Output Format:
	For each test case, output YES if Amala can achieve exactly Y marks, NO otherwise.
	Input: Output:
	5
	1 8 4 NO
	3 6 12 YES
	4 5 0 YES
	10 10 100 YES
	8 5 36 NO
	Solution:
	#include <stdio.h>
	int main(void) {
	int t,n,x,y;
	scanf("%d",&t);
	while(t--)
	{
	scanf("%d %d %d",&n,&x,&y);
	if(y<x&&y!=0)
	printf("No\n");
	else if(y==0)
	printf("Yes\n");
	else if(y%x==0)
	printf("Yes\n");
	else
	printf("No\n");
	}
	return 0;
	}
	
	(v) Problem:
		5	**3**	**2**	PO2	**2.5.2**
						
						5
							1	**2**	PO2	**2.5.2**

1b	Alice, Bob and Charlie are bidding for an artifact at an auction. Alice bids A rupees, Bob bids B rupees, and Charlie bids C rupees (where A, B, and C are distinct). According to the rules of the auction, the person who bids the highest amount will win the auction. Determine who will win the auction.
	Input Format:
	• Each test case contains three integers A, B, and C, — the amount bid by Alice, Bob, and Charlie respectively.
	Output Format:
	• For each test case, output who (out of Alice, Bob, and Charlie) will win the auction.
	Solution:
	#include <stdio.h>
	int main(void) {
	int t,a,b,c;
	scanf("%d",&t);
	for(int i=0;i<t;i++){
	scanf("%d %d %d",&a,&b,&c);
	if(a>b && a>c){
	printf("Alice\n");
	}
	else if(b>c){
	printf("Bob\n");
	}
	else{
	printf("Charlie\n");
	}
	}
	return 0;
	}
	(OR)
	(i) Problem:
	Most programmers will tell you that one of the ways to improve your performance in competitive programming is to practice a lot of problems. Naveen decides to solve at least 10 problems every week for 4 weeks. Given the number of problems he actually solved in each week over 4 weeks as P1,P2,P3, and P4, output the number of weeks in which he met his target.
	Input Format:
	• There is a single line of input, with 4 integers P1, P2,P3, and P4. These are the number of problems solved by Naveen in each of the 4 weeks.
	Output Format:
	Output a single integer in a single line - the number of weeks in which Naveen solved at least 10 problems.
	Input: Output:
	12 15 8 10 3
	Solution:
	#include <stdio.h>
	int main(void) {
	// your code goes here
	int arr[4],i,count=0;
	for(i=0;i<4;i++)
	{
	scanf("%d",&arr[i]);
	}
	for(i=0;i<4;i++)
	{
	if(arr[i]>=10)
	count++;
	else
	continue;
		5
		
		5
			2
			
			3
				2
				
				2
					PO2
					
					PO2
						2.5.2
						
						2.5.2

	}
	printf("%d",count);
	}
	(ii) Problem:
	Nakul wants to go to a music concert with his friends. The seats for the audience is arranged in descending order of the maximum number of tickets booked for a single booking. As per the seating arrangement booking with the highest number of people is allotted the seats in the front rows and the with the lowest number of people is allotted the seats in the last row. The event management firm has planned to develop the software which displays the exact seat layout if the total number of rows is provided. Can you help them with the logic of doing so?
	Input Format:
	The only line of input has a single integer representing the total number of rows for the concert.
	Output Format:
	Print the seating arrangement layout based on the number of rows provided.
	`Enter the number of rows: 7`
	`7 7 7 7 7 7 7`
	`6 6 6 6 6 6`
	`5 5 5 5 5`
	`4 4 4 4`
	`3 3 3`
	`2 2`
	`1`
	Solution:
	#include <stdio.h>
	int main()
	{
	int i, j, rows;
	printf("Enter the number of rows: ");
	scanf("%d", &rows);
	for (i = rows; i >= 1; --i) {
	for (j = 1; j <= i; ++j) {
	printf("%d\t",rows);
	}
	printf("\n");
	}
	return 0;
	}
	(iii) Problem:
	Vasya likes the number 239. Therefore, he considers a number pretty if its last digit is 2, 3 or 9. Vasya wants to watch the numbers between L and R (both inclusive), so he asked you to determine how many pretty numbers are in this range. Can you help him?
	Input Format:
	Each test case contains two space-separated integers L and R.
	Output Format:
	For each test case, print a single line containing one integer — the number of pretty numbers between L and R.
	Input Output
	`2` `3`
	`1 10` `8`
	`11 33`
	Solution:
	int main(void) {
	int t,j,count=0,i;
	long int l,r;
		5	**2**	**2**	PO2	**2.5.2**

```
    scanf("%d",&t);
    for(i=0;i<t;i++)
    {
       scanf("\n%ld %ld",&l,&r);
       for(j=l;j<=r;j++)
       {
          if((j%10==2)||(j%10==3)||(j%10==9))
          count++;
       }
       printf("\n%d",count);
       count=0;
    }
          return 0;
}
```

		5	**3**	**2**	PO2	**2.5.2**

(iv) Problem:

Shreyansh likes to play with arrays. He started solving many problems related to array and after a while he thought that he had become an expert in this field and could solve any problem related to array. Seeing this shubham decided to challenge him with a problem related to array. Shubham gave him the following problem.

- Given an array A[] with N integers and count the total number of prime numbers in the array.

Solution:

```
#include<stdio.h>
int main()
{
        int minimum, maximum, flag, count=0, i, j;
        printf("Enter minimum number: ");
        scanf("%d", &minimum);
        printf("Enter maximum number: ");
        scanf("%d", &maximum);

        /* Generating and counting prime numbers */
        for(i=minimum; i<=maximum; i++)
        {
                flag = 0;
                for(j=2; j <= i/2; j++)
                {
                        if(i%j==0)
                        {
                                flag=1;
                                break;
                        }
                }
                if(flag==0 && i>=2)
                {
                        printf("%d\t",i);
                        count++;
                }
        }
        printf("\n Prime Count = %d", count);
        return(0);
}
```

		5	**3**	**2**	PO2	**2.5.2**

(v) Problem:

Arun is a professor and he wants to find who has secured first mark and second mark in his subject. Help him to find the same.

Input Format:
Get N students marks in his class
Output Format:
print the first mark and second mark
Solution:

```
#include <stdio.h>
int main() {
  int array[10] = {101, 11, 3, 4, 50, 69, 7, 8, 9, 0};
  int loop, largest, second;
  if(array[0] > array[1]) {
    largest = array[0];
    second  = array[1];
  } else {
    largest = array[1];
    second  = array[0];
  }
  for(loop = 2; loop < 10; loop++) {
    if( largest < array[loop] ) {
      second = largest;
      largest = array[loop];
    } else if( second < array[loop] ) {
      second =  array[loop];
    }
  }
  printf("Largest - %d \nSecond - %d \n", largest, second);
  return 0;
}
```

Alternatively, may be done by finding minimum number twice.

Part – B
(1 x 25 = 25 Marks)

Instructions: This section has only ONE question with internal choice.

2a **(i) Problem:**
Neha has a string S with her fname and lname. She needs to know the number of characters in fname and lname. Help her to find the length of fname and lname. After finding the length she needs to combine the fname and lname. Help her to implement the logic.
Input format:
Get the first name and last name
Output:
Display the number of characters and combined string
Solution:

```
#include<stdio.h>
#include<string.h>
void main()
{       clrscr();
        char s1[25],s2[25];
        int l;
        printf("Enter first string=");
        scanf("%s",s1);
        printf("Enter second string=");
        scanf("%s",s2);
        strcat(s1,s2);
        l=strlen(s1);
        printf("Concatenate string=%s\n",s1);
        printf("Length of concatenate string=%d",l);
        getch();
}
```

5	**1**	**3**	**PO1**	**1.7.1**

	(ii) Problem:
	Chan has a string S with him. He is happy if the string contains a contiguous substring of length strictly greater than 2 in which all its characters are vowels. Determine whether Chan is happy or not.
	Input Format:
	Get single line of input, a string S.
	Output Format:
	if Chan is happy, print HAPPY else print SAD.
	Solution:
	`#include <stdio.h>`
	`#include<string.h>`
	`int main(void)`
	`{`
	`int t;`
	`scanf("%d",&t);`
	`char s[1000];`
	`while(t--)`
	`{`
	`int i,count=0;`
	`scanf("%s",s);`
	`for(i=0;i<strlen(s)-1;i++)`
	`{`
	`if(s[i]=='a'||s[i]=='e'||s[i]=='i'||s[i]=='o'||s[i]=='u')`
	`{`
	`count++;`
	`}`
	`else`
	`{`
	`if(count>2)`
	`break;`
	`else`
	`count=0;`
	`}`
	`}`
	`if(count>2)`
	`{`
	`printf("Happy\n");`
	`}`
	`else`
	`{`
	`printf("Sad\n");`
	`count=0;`
	`}`
	`}`
	`return 0;`
	`}`
	Or alternatively can be done by checking if both character i and character i+1 are vowels in the same if condition.
		5	**2**	**3**	**PO2**	**2.5.2**

	(iii) Problem: Charvy invented a modified wordle. There is a hidden word S and a guess word *T*, both of equal length. Charvy defines a string M to determine the correctness of the guess word. For the *ith* index: • If the character at the *ith* index is same both in S and T then the ith character of M is G otherwise B. • Given the hidden word *S* and guess *T*, determine string M. **Input Format** • Each test case contains of two lines of input. • First line contains the string S - the hidden word. • Second line contains the string T - the guess word.	**5**	**1**	**3**	**PO2**	**2.5.2**

Output Format
For each test case, print the value of string M

```
1
cat
rat
BGG
```

Solution:

```
#include <stdio.h>

int main() {

        int t, x, y, z,n1;
        char s[n1], r[n1];

        scanf("%d", &t);
        scanf("%d",&n1);

        while ( t-- ){

           scanf("%s %s", &s, &r);

           for(x = 0; x < n1; x++)
           {
              if (s[x] == r[x])
              {
                 printf("G");
              }else{
                 printf("B");
              }
           }
           printf("\n");
        }
        return 0;
}
```

5	**2**	**3**	**PO2**	**2.5.2**	

(iv) Problem:
Amir has a string *S* with him. He needs to know the occurrence of a particular character and also number of occurrences of that character. Help him to find the logic for the same.
Input format:
Get the input string and a character to check occurrence
Output:
Total number of occurrences of the given character.

```
#include <string.h>
int main()
{
   char s[1000],c;
   int i,count=0;
    printf("Enter  the string : ");
   gets(s);
   printf("Enter character to be searched: ");
   c=getchar();
    for(i=0;s[i];i++)
   {
           if(s[i]==c)
           {
      count++;
                   }
           }
           printf("character '%c' occurs %d times \n ",c,count);
           return 0;
```

2b	}
	
	(v) Problem:
	Meena has a string *S* with combination of upper case and lower case characters of length N. She needs to convert it to lower case. Help her do the same.
	Input Format
	Enter the string: SRMist
	Output Format
	Given string : SRMist
	Lower case : srmist
	#include<stdio.h>
	void main(){
	int i;
	char string[40];
	printf("Enter the string : ");
	gets(string);
	//For loop to read each alphabet//
	for(i=0;string[i]!='\0';i++){
	if(string[i]>=65&&string[i]<=90){
	string[i]=string[i]+32;
	}
	}
	printf("The converted lower case string is : ");
	puts(string);
	}
	(OR)
	(i) Problem:
	Shivam is the youngest programmer in the world, he is just 12 years old. Shivam is learning programming and today he is writing basic program. The task is very simple: given two integers A and B, write a program to add these two numbers ***using pointers*** and output it.
	Input Format:
	Get two integers
	Output Format:
	Sum of two numbers
	Solution:
	#include <stdio.h>
	int main()
	{
	int fno, sno, *ptr, *qtr, sum;
	printf("\n\n Pointer : Add two numbers :\n");
	printf(" Input the first number : ");
	scanf("%d", &fno);
	printf(" Input the second number : ");
	scanf("%d", &sno);
	ptr = &fno;
	qtr = &sno;
	sum = *ptr + *qtr;
	printf(" The sum of the entered numbers is : %d\n\n",sum);
	return 0;
	}
	(ii) Problem:
	Yogesh got his three subject marks Maths, Physics and Chemistry from college. Now he want to find out in which subject he get highest *marks*. The marks in Maths is A , physics is B and chemistry is C. Use ***pointers*** to implement the logic.
	Input Format
	• Input contains A , B , C i.e. Marks in maths , physics and chemistry respectively.
	Output Format
	The subject in which he gets highest marks.
		5
		
		5
		
		5
			3
			
			1
			
			1
				3
				
				4
				
				4
					PO2
					
					PO1
					
					PO2
						2.5.2
						
						1.7.1
						
						2.5.2

	Solution:
	#include <stdio.h>
	int main()
	{
	int mat,phy,che, *ptr1=&mat,*ptr2=&phy,*ptr3=&che;
	scanf("%d", ptr1);
	scanf("%d", ptr2);
	scanf("%d", ptr3);
	
	if(*ptr1>*ptr2 && *ptr1>*ptr3)
	{
	printf("Maths");
	}
	else if(*ptr2>*ptr3 && *ptr2>*ptr3)
	{
	printf("Physics");
	}
	else
	printf("Chemistry");
	return 0;
	}
	(iii) Problem
	Ravi has to travel to another place. For this, he can avail any one of two cab services.
	• The first cab service charges X rupees.
	• The second cab service charges Y rupees.
	Ravi wants to spend the **minimum** amount of money. Which cab service should Ravi take? Use ***pointers*** to implement the concept
	Input Format
	• Each test case contains two integers X and Y- the prices of first and second cab services respectively.
	Output Format
	For each test case, output FIRST if the first cab service is cheaper, output SECOND if the second cab service is cheaper, output ANY if both cab services have the same price.
	3
	20 60
	First
	40 40
	Any
	50 30
	Second
	Solution:
	#include <stdio.h>
	#include <stdlib.h>
	void main()
	{
	int n, fno,sno,*ptr1=&fno,*ptr2=&sno;
	scanf("%d", &n);
	for (int i = 0; i < n; i++)
	{
	scanf("%d", ptr1);
	scanf("%d", ptr2);
	if(*ptr1<*ptr2)
	{
	printf("\nFirst");
	}
	else if(*ptr2<*ptr1)
	{
	printf("\nSecond");
	}
	else
	printf("\nAny");
		5	**2**	**4**	**PO2**	**2.5.2**

	}
	}
	(iv) Problem
	Bunty is a 10 year old boy playing a game of swap the two numbers A & B and write them on a board in correct order. He is in confusion of doing this. So help him by taking this two number A and B as input and after Swapping print the output on screen using ***pointers and functions.***
	Input Format
	Input line contains two numbers A and B.
	Output Format
	Output in a single line the two numbers after swap them.
	Solution:
	#include <stdio.h>
	void swapNumbers(int *x,int *y);
	int main()
	{
	int e1,e2;
	scanf("%d",&e1);
	scanf("%d",&e2);
	swapNumbers(&e1,&e2);
	printf("%d %d",e1,e2);
	return 0;
	}
	void swapNumbers(int *x,int *y)
	{
	int tmp;
	tmp=*x;
	*x=*y;
	*y=tmp;
	}
		5	**2**	**4**	**PO2**	**2.5.2**

	(v) Problem:
	Alice and Bob are very good friends and they always distribute all the eatables equally among themselves. Alice has A chocolates and Bob has B chocolates. Determine whether Alice and Bob can distribute all the chocolates equally among themselves. Implement the logic using ***functions***
	Note that: It is not allowed to break a chocolate into more than one piece. No chocolate shall be left in the distribution.
	Input Format:
	Each test case contains two space-separated integers A and B, the number of chocolates that Alice and Bob have, respectively.
	Output Format:
	For each test case, output on a new line YES if Alice and Bob can distribute all the chocolates equally, else output NO.
	Solution:
	#include <stdio.h>
	void solve()
	{
	int a, b;
	scanf("%d%d", &a, &b);
	
	((a + b) % 2 == 0) ? printf("YES\n") : printf("NO\n");
	}
	int main(void) {
	int T;
	scanf("%d", &T);
	while(T--){
	solve();
	}
	return 0;
	}
		5	**3**	**4**	**PO2**	**2.5.2**