C Programming Module 2 Part A Solutions

@ Nishant and Ujjwal

1. List the control structures in C.

Ans: There are two types of Control Structures in C –

(a) Conditional Control (decision making) – ex: if, else, if-else statements, switch.
(b) Repetitive Control (looping) – ex: for, while, do-while loops.

2. List the decision making statements in C.

Ans: if, if-else, nested- if, switch case.

3. List the loop control statements in C.

Ans: for, while and do-while(conditions are mentioned after the process is executed).

4. Explain Continue statement in C.

Ans: The ‘continue’ is a C reserved word or keyword that is used to skip the current iteration of loop. Any statements below the continue is skipped for that iteration!

5. Explain goto statement in C.

Ans: The goto statement is used to jump the control to a particular line of code!

Syntax: goto Label;

Before you use this statement, make sure to mark a ‘Label’ block in your code. For example:

```
int main()
{
printf("Statement in normal flow!\n");
goto Label;
printf("This statement doesn't execute because the compiler now skips this line and goes to Label block!");
Label:
   printf("Statement inside Label block!!\n");
return 0;
}
```

Output:

Statement in normal flow!

Statement inside Label block!!

6. Explain break statement in C.

Ans. Break Statement makes the control jump out of the loop it is in. The remaining iterations of the loop won't execute! It is important that the break statement be included within the loop for execution.

7. Compare the difference between entry controlled and exit controlled statements.

Entry controlled statements – check the condition first and then executes the body

The body doesn't execute if the condition is false. For and While loops follow entry controlled statements for execution.

Exit controlled statements – executes the body first and then checks the condition.

The body executes once if the condition is false. A do-while loop follows a similar implementation.

8. Explain switch statement in C.

Ans. The switch statement allows us to execute one code block among many alternatives depending upon the value of the expression. If no case matches the value of expression, default case executes.

Syntax:

```
switch (expression)
  {
  case constant1:
    // statements
    break;
  case constant2:
    // statements
    Break;
  default:
    // default statements
   }
```

9. Find the output of the following code:

```
int main()
{
 int a = 1, b = 2, c = 3, d = 4, e;
 if(e= (a & b | c ^ d))
 printf("%d", e);
 return 0;
}
```

Output: 7

10. Find the output of the following code:

```
void main()
{
```

```
char c = 125;
do
printf("\n%d", c);
while(c++);
}
```

Output: 125
126
127
-128
-127
-126
-125

And so on till the value reaches 0

11. Find the output:

```
void main()
{
 for(;;)  {
 printf("%d", 10);
 }
}
```

Soln: for(;;) -> forms an infinite loop, so the program keeps executing

12. Find the output of the following code.

```
void main()
{
 printf("hi!");
 if (!0)
 printf("bye");
}
Output: hi!bye
```

13. Find the output of the following code:

```
void main()
{
int a =1;
if(a)
printf("test");
else ;
printf("again");
}
```

Output: testagain

14. Find the output of the following code:

```
int main()
{
int a;
```

```
for(a = 5; --a;)
printf("\n%d", a);
return 0;
}
```

Output:

4
3
2
1

15. Find the output of the following code:

```
void main()
{
float i;
for(i = 0.1;i < 0.4; i += 0.1)
printf("%.1f\n", i);
}
```

Output:

0.1
0.2
0.3

16.Explain switch case execution process with and without break statement.

Ans. When the expression value in switch() matches the value of a case, the statements inside it execute and when break statement is encountered, the control exits the switch block.
If break statement is not used, all the remaining cases are executed till break statement is encountered.

17. Find the output of the following code:

```
void main()
{
int i = 3;
for(i--; i< 7; i = 7)
printf("%d", i++);
}
```

Ans. 7

Hint: Follow the syntax of for-loop!

18. Find the output of the following code:

```
int main()
{
int i = 1;
for(; i< 4; i++);
printf("%d", i);
return 0;
}
```

Ans. 4

Remember to check the code thoroughly and look for semi-colons after the for-loop!

19. Write a program in C to display the n terms of odd natural number and their sum.

Soln:

```
#include <stdio.h>
int main(){
int n_terms,i=1,counter =0, sum = 0;
printf("Input Number of Terms: ");
scanf("%d",&n_terms);
printf("The Odd Numbers are:");
while(counter<n_terms){
    printf("% d",i);
    sum += i;
    i+= 2;
    counter ++;
  }
  printf("\nThe Sum of Odd Natural Numbers upto %d terms: %d",n_terms,sum);
  return 0;
  }
```

20. Write a program in C to display the multiplication table of a given integer.

Soln:

```
#include <stdio.h>
int main(){
int number,i;
printf("Input the number(Table to be calculated): ");
scanf("%d",&number);
for(i=1;i<=10;i++){
    printf("%d X %d = %d\n",number,i,number*i);
}
return 0;
}
```

C Programming Module 2 Part B Solutions

@ Bharath and Ujjwal

1. Write a program in C to display the n terms of square natural number and their sum.1 4 9 16 ... n Terms

Sample Input/ Output:

Input the number of terms : 5

Expected Output :

The square natural upto 5 terms are :1 4 9 16 25

The Sum of Square Natural Number upto 5 terms = 55

```
#include <stdio.h>
int main(void) {
	int n;
	int x = 1, i = 0, sum = 0;
	printf("Input the number of terms:");
	scanf("%d", &n);
	printf("The squares of natural numbers upto %d terms are: ", n);
	for (i; i < n; i++) {
		printf("%d ", x * x);
		sum += x * x;
		x++;
	}
	printf("\nThe sum of square natural number upto %d terms = %d", n,
sum);
	return 0;
}
```

---

2. Write a program in C to find the prime numbers within a range of numbers.

Sample Input/ Output:

Input starting number of ranges: 1

Input ending number of range: 50

Expected Output:

The prime number between 1 and 50 are: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

```
#include <stdio.h>
int main(void) {
	int x,y;
	printf("Input starting number of range: ");
	scanf("%d", &x);
	printf("Input ending number of range: ");
	scanf("%d", &y);
	printf("The prime numbers between %d and %d are:", x, y);
	int i, prime = 1;
	for (x; x<=y; x++){
		prime = 1;
		if (x == 1) prime=0;
		for (i = 2; i<x; i++){
			if (x%i==0){
				prime = 0;
				break;
			}
		}
		if (prime) printf(" %d", x);
	}
	return 0;
}
```

---

3. Write a C program to display the traffic control signal lights based on the following.

i. If user entered character is R or r then print RED Light Please STOP.

ii. If user entered character is Y or y then print YELLOW Light Please Check and Go.

iii. If user entered character is G or g then print GREEN Light Please GO.

iv. If user entered some other character then print THERE IS NOSIGNAL POINT.

```
#include <stdio.h>
int main(void) {
	char n;
	scanf("%c", &n);
	if (n=='R' || n=='r') printf("RED Light Please STOP.");
	else if (n=='Y' || n=='y') printf("YELLOW Light Please Check and Go");
	else if (n=='G' || n=='g') printf("GREEN Light Please GO.");
	else printf("THERE IS NOSIGNAL POINT");
}
```

---

4. Admission to a professional course is subject to the following conditions:

i. Marks in Mathematics >=60

ii. Marks in Physics >= 50 Marks in Chemistry >=40

iii. Total in all three subjects >=200

iv. Total in Mathematics and Physics >=150

Given the marks in the three subjects, Write a C program to process the application to list the eligible candidates

```
#include <stdio.h>
int main(void) {
	int M, P, C;
	scanf("%d %d %d", &M, &P, &C);
	if (M>=6 && P>=50 && C>=40 && M+P+C>=200 && M+P>=150) printf("eligible");
	else printf("not eligible"); }
```

---

5. Write a C program to compute the real roots of a quadratic equation ax2 + bx+ c = 0. The program should request for the values of the constants a, b and c and print the values of x1 and x2.

Use the following rules:

i. No solution, if both a and b are zero There is only one root, ifa=0

ii. There are no real roots, if b2 -4ac is negative Otherwise, there are two real roots

Write a C program to test all the above conditions

```
#include <stdio.h>
#include <math.h>
int main(void) {
	double a,b,c;
	scanf("%lf %lf %lf", &a, &b, &c);
	double det = b*b-4*a*c;
	if (a==0 && b==0){
		printf("No solution");
	}
	else if (a==0){
		printf("%lf", -c/b);
	}
	else if (det<0) {
		printf("No real roots");
	}
	else{
		printf("%lf", (-b+sqrt(det))/2*a);
		printf("%lf", (-b-sqrt(det))/2*a);
	}
	return 0;
}
```

---

6. Write a program that counts from one to ten, prints the values on a separate line for each, and includes a message of your choice when the count is 3 and a different message when the count is 7

```
#include <stdio.h>
int main(void){
	int i;
	for (i=1; i<11; i++){
		printf("\n%d ", i);
		if (i==3) printf("Hello");
		if (i==7) printf("World");
	}
	return 0;
}
```

---

7. Write a C program to calculate commission for the input value of sales amount. Commission is calculated as per the following rules:

i. Commission is nil for sales amount Rs5000/.

ii. Commission is 2% for sales when sales amount is greater than 5000and less than equal to 10000.

iii. Commission is 5% for sales amount greater than 10000.

```
#include <stdio.h>
int main(void){
	double sales;
	scanf("%lf", &sales);
	double commission;
	if (sales<=5000) {
		commission = 0;
	}
	else if (sales>5000 && sales<=10000) {
		commission = 0.02*sales;
	}
	else commission = 0.05*sales;
	printf("%.2lf", commission);
```

```
	return 0;
}
```

---

8. A character is entered through keyboard. Write a C program to determine whether the character entered is a capital letter, a small case letter, a digit or a special symbol using if-else and switch case. The following table shows the range of ASCII values for various characters.

Characters ASCII values

A–Z 65 –90

a–z 97 –122

0–9 48 – 57

Special symbols 0 – 47, 58 – 64, 91 – 96, 123 - 127

```
#include <stdio.h>
int main(void) {
	int a;
	scanf("%c", &a);
	if (a>=65 && a<=90) printf("Capital Letter");
	else if (a>=97 && a<=122) printf("Lowercase Letter");
	else if (a>=48 && a<=57) printf("Number");
	else printf("Special character");
	return 0;
}
```

---

9. If cost price and selling price of an item S input through the keyboard, write a program to determine whether the seller has made profit or incurred loss.Write a C program to determine how much profit or loss incurred in percentage.

```
#include <stdio.h>
int main(void) {
	double cp, sp;
```

```
	scanf("%lf %lf", &cp, &sp);
	if (cp>sp){
		printf("You have an incurred a loss of %.2lf", cp-sp);
		printf("\nLoss percentage: %.2lf", ((cp-sp)/cp)*100);
	}
	else {
		printf("You have gained a profit of %.2lf", sp-cp);
		printf("\nProfit percentage: %.2lf", ((sp-cp)/cp)*100);
	}
	return 0;
}
```

---

10. Write a C program to print the following pattern.

1

3 5

7 9 11

13 15 17 19

```
#include <stdio.h>
int main(void) {
	int n, x=1, i, u;
	printf("How many rows do you want: ");
	scanf("%d", &n);
	//To find the largest possible number for this sequence
	for (i=0; i<n; i++){
		for (u=0; u<=i; u++) {
			x+=2;
		}
	}
	//To find the number of digits of the largest number
	int digits = 1, spaces = 1, y;
```

```
	while (x/10!=0){
		x=x/10;
		digits++;
	}
	x=1;
	//To actually print the sequence
	for (i=0; i<n; i++){
		for (u=0; u<=i; u++) {
			printf("%d ", x);
			y=x;
			spaces = 1;
			//To find the appropriate number of blank space characters
			for (y; y/10!=0; y=y/10){
				spaces++;
			}
			//To print blank spaces
			for (spaces; spaces<digits; spaces++) {
				printf(" ");
			}
			x+=2;
		}
		printf("\n");
	}
	return 0;
}
```

---

11. Write a C program to print the following pattern.

1

12

123

1234

```
#include <stdio.h>

int main(void) {

	int n, i, x;

	scanf("%d", &n);

	for (i=1; i<=n; i++) {

		for (x=1; x<=i; x++) {

			printf("%d", x);

		}

		printf("\n");

	}

}
```

---

12. Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression1+x+x2+x3+...xn.

For example: if n is 3 and x is 5, then the program computes 1+5+25+125. Print x, n, the sum. Perform error checking. For example the formula does not make sense for negative Exponents – if n is less than 0.

Have your program print an error message if n<0,then go back and read in then pair of numbers of without computing the sum. Are any values of x also illegal? If so, test for them too.

```
#include <stdio.h>

int main(void) {

	int a = 1, i, sum = 1;

	double x, n;

	progression:

	scanf("%lf %lf", &n, &x);

	if (n-(int)n!=0){

		printf("Please enter a natural number for n\n");

		goto progression;

	}

	for (i = 0; i<n; i++) {

		a*=x;
```

```
            sum+=a;
        }
        if (n<0) {
            printf("Please enter a positive number for n\n");
            goto progression;
        }
        printf("%d",sum);
        return 0;

}
```

---

13. Write a C program to print Armstrong numbers between 1 to n where n value is entered by the user.

[Hint: Armstrong number is defined as the sum of cubes of individual digits of a number. e.g. 371 = 33 + 73 + 13]

```
#include<stdio.h>
int main()
{
    int i,I,n,r,s=0;
    printf("Enter a number:");
    scanf("%d",&n);
    for (i=1;i<=n;i++)
    {
            I=i;
            s=0;
            r=0;
        while (i!=0)
        {
            r=i%10;
            s=s+(r*r*r);
            i=i/10;
```

```
        }
        i=I;
        if (s==I)
        printf("%d\n",I);
    }
}
```

---

14. Write a C program to generate all prime numbers between 1 and n, where n value is supplied by the user.

```
#include <stdio.h>
int main(void) {
       int n, i, j, isPrime = 1;
       scanf("%d", &n);
       for (i=2; i<=n; i++){
              for (j=2; j<i; j++){
                     if (i%j==0) isPrime = 0;
              }
              if (isPrime) printf("%d ", i);
              isPrime = 1;
       }
       return 0;
}
```

---

15. Write a C program to print first n lines of Floyd's Triangle.

1

2 3

4 5 6

7 8 9 10

```c
#include <stdio.h>
int main(void) {
	int n, x=1, i, u;
	printf("How many rows do you want: ");
	scanf("%d", &n);
	//To find the largest possible number for this sequence
	for (i=0; i<n; i++){
		for (u=0; u<=i; u++) {
			x+=1;
		}
	}
	//To find the number of digits of the largest number
	int digits = 1, spaces = 1, y;
	while (x/10!=0){
		x=x/10;
		digits++;
	}
	x=1;
	//To actually print the sequence
	for (i=0; i<n; i++){
		for (u=0; u<=i; u++) {
			printf("%d ", x);
			y=x;
			spaces = 1;
			//To find the appropriate number of blank space characters
			for (y; y/10!=0; y=y/10){
				spaces++;
			}
			//To print blank spaces
			for (spaces; spaces<digits; spaces++) {
				printf(" ");
			}
```

```
                    x++;
                }
                printf("\n");
        }
        return 0;
}
```

---

16. Write a C program to print the following series 1/1! + 2/2! + 3/3! + ...

```
int main()
{
    int num = 1, count;
    float sum = 0.0, fact;
    while(num <= 7)  {
        fact = 1;
        for(count = 1; count <= num; count++)  {
            fact = fact * count;
        }
        sum = sum + (num / fact);
        num++;
    }
    printf("Sum of series is %f\n", sum);
    return 0;
}
```

---

17. Write a C program to compute and display the sum of all integers that are divisible by 6 but not divisible by 4 and lie between 0 and 100. The program should also count and display the number of such values.

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

```
    int i,sum,count;
    sum=0;
    count=0;
    for (i=0;i<=101;i+=6) {
        if (i%4!=0) {
            sum+=i;
            count+=1;
        }
    }
    printf("Sum of such digits: %d\n", sum);
    printf("Count of such digits: %d", count);
}
```

---

18. Write a C program to find the LCM and GCD of two integers.

```
#include < stdio.h >
int main()
{
    int num1, num2, gcd, lcm, count = 1, small;
    printf("Enter 2 integer numbers\n");
    scanf("%d%d", &num1, &num2);
    small = (num1 < num2) ? num1 : num2;
    while(count <= small)  {
        if(num1 % count == 0 && num2 % count == 0)  {
            gcd = count;   }
        count++;
    }
    lcm = ( num1 * num2 ) / gcd;
    printf("GCD = %d\nLCM = %d\n", gcd, lcm);
    return 0;  }
```

---

19. Write a program in C to display the n terms of odd natural number and their sum.

```
#include <stdio.h>
int main() {
    int n,i,sum;
    sum=0;
    scanf("%d",&n);
    for (i=1;i<=n;i+=2) {
        printf("%d ",i);
        sum+=i;
        }
    printf("\nSum is: %d", sum);
    }
```

---

20. Write a C program to print the following pattern.

```
#include <stdio.h>
int main() {
    int i,j;
    for (i=1;i<=4;i++) {
        for (j=1;j<=i;j++) {
            printf("%d",i);
        }
        printf("\n");
    }
}
```

C Programming Module 2 Part C Solutions

@Nishant

1. Predict the output of the following code:

```
void main() {
int x=4;
if(x=4) {
if (x=4)
break;
printf("HI");
}
printf("BYE");
}
```

Soln. This code gives an error. Since, **break can only be used inside loops.**

2. Predict the output of the following code.

```
int main() {
     int i = 1024;
     for (; i; i >>= 1)
          printf("IARE");
     return 0;
}
```

Soln. On close inspection of code, i = 1024 and the update expression applies bitwise right shift operation as i = i >> 1.

Thus, the **output will be**:

IAREIAREIAREIAREIAREIAREIAREIAREIAREIAREIARE **(The loop runs 11 times)**

3. Predict the output of the following code.

```
int main() {
  int i = 5, j = 10, k = 1;
  if(++i || ++j ) {
       k = i + j;}
  else {
       k = i - j; }
```

```
printf("%3d%3d%3d", i, j, k);
return 0;
 }
```

Soln. Since we have an '||' (or) operator in between ++i and ++j, the program only increases the value of the first variable i.e., 'i' and so the **Output is:**
6 10 16 *(%3d adds a space before the integer)*

4. Predict the output:

```
int main()    {
int i = -5;
while(i<=5) {
if(i>=0)
 break;
else {
 i++;
 continue;
}
printf("IARE");
}
return 0;
}
```

Soln. Since **continue** keyword is used, printf never executes and so we get a blank screen as output.

5. Predict the output:

```
int main()    {
int i = 3;
while (i--) {
 int i = 100;
 i--;
 printf("%d ", i);
}
return 0;
}
```

Soln. This question tests our understanding of **Global and local declarations**!
Outside the while loop we have i=3 -> global declaration. Inside the while loop we have i = 100 -> Local declaration. And thus, the while loop runs **3 times**, giving us the
Output: 99 99 99

6. Predict the Output:

```
int main()    {
int i;
goto LOOP;
for(i = 0 ; i< 10 ; i++)    {
printf("IARE\n");
LOOP: break;
}
return 0;
}
```

Soln. When the compiler enters the main() function -> it encounters the goto statement and thus goes to -> LOOP and there's a break statement inside it -> so the for loop breaks and **No Output is displayed**

7. Predict the Output of the following code:

```
int main()    {
unsigned short int i = 65000;
while(i++ != 0);
printf("ans : %d", i);
return 0;
}
```

Soln. unsigned allows only positive values and range of short is 65,535 and so the while loop (even though it is a null loop) runs till i becomes 65,535 then i becomes 0, but because we have an i++ , the value of i becomes 1.
Output is: ans : 1

8. Predict the output of the following code:

```
int main()    {
int i = 65;
char j='A';
while(i< j);
 printf(" %d", (i ^ j )<< 2);
return 0;
 }
```

Soln. While loop is null loop. So printf is executed and 65 and A are converted to its binary equivalents and bitwise XOR (^) is applied. And we get the **Output: 0.**

9. Predict the output:

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

```
 for(i=65; i<(65+26);i++)
 printf("%c ",i);
return 0;
}
```

Soln. %c in printf converts the integer into its respective character following ASCII standards.
And so the **Output is: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z**

10. Predict the output:

```
void main()
{
int i, j, k;
for(i = 1;i < 3; i++)
{
    for( j = 1; j < 3; j++)
    {
        for(k = 1; k < 3; k++)
        {
            if(j == k)
                break;
            else
            {
            printf("%d\t%d\t%d\n", i,j, k);
            continue;
            }
        }
    }
 }
}
```

Soln. On proper tracing, the **output is:**
1 2 1
2 2 1
(\t means tab space)