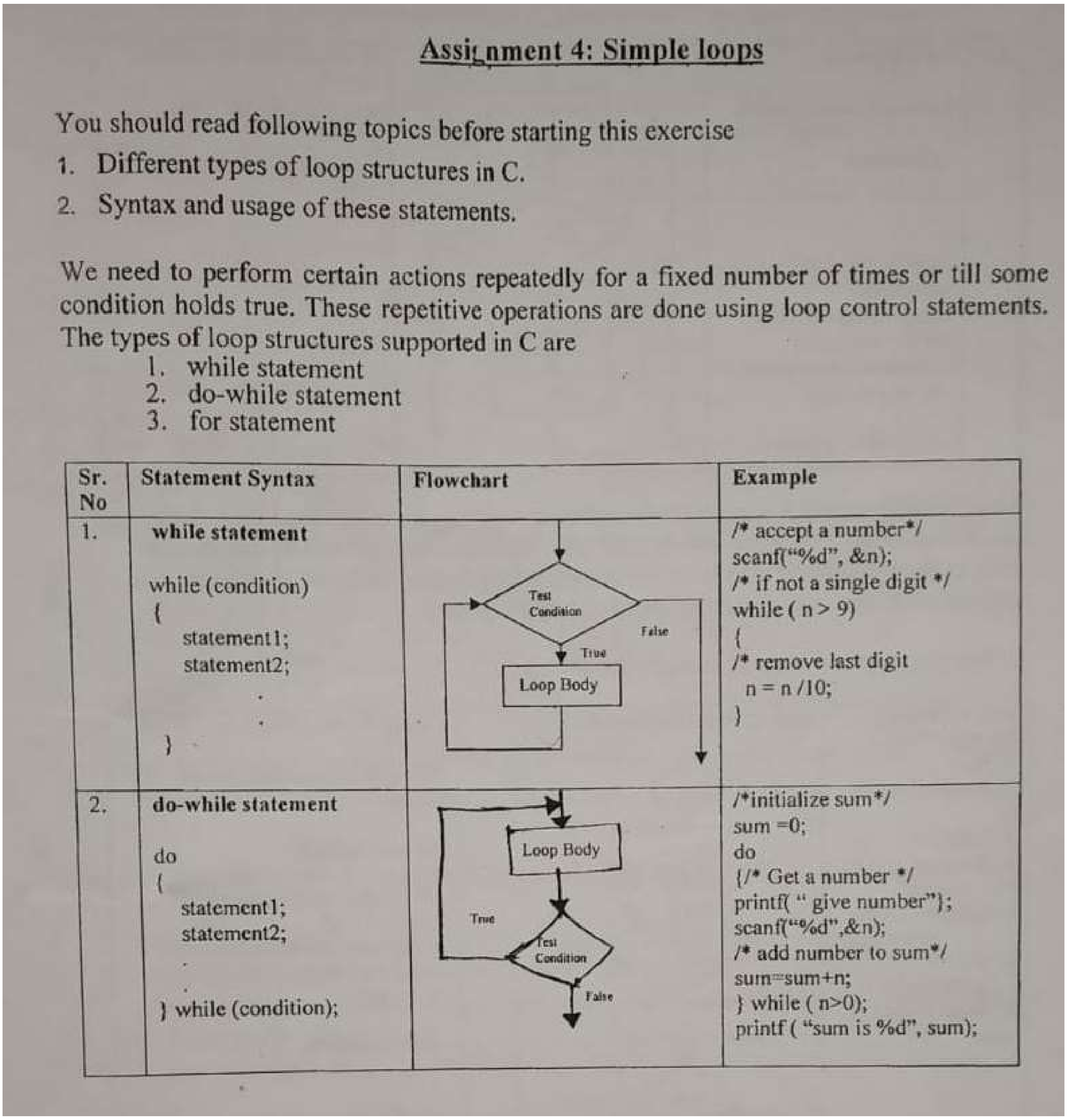
BCA 115: LAB I (C Programming)

Assignment 4



Assignment 4: Simple loops

You should read following topics before starting this exercise

1. Different types of loop structures in C.
2. Syntax and usage of these statements.

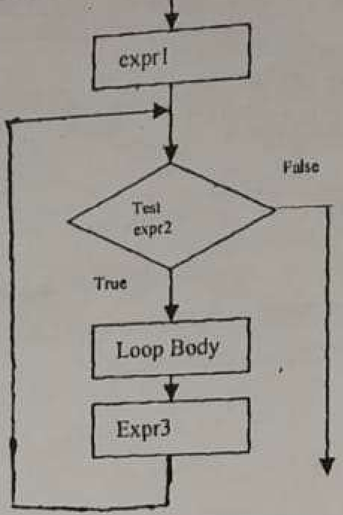
We need to perform certain actions repeatedly for a fixed number of times or till some condition holds true. These repetitive operations are done using loop control statements. The types of loop structures supported in C are

1. while statement
2. do-while statement
3. for statement

Sr. No	Statement Syntax	Flowchart	Example

1.	**while statement**
	while (condition)
	{
	statement1;
	statement2;
	.
	.
	}
		Test Condition
		False
		True
		Loop Body
			/* accept a number*/
			scanf("%d", &n);
			/* if not a single digit */
			while (n > 9)
			{
			/* remove last digit
			n = n /10;
			}

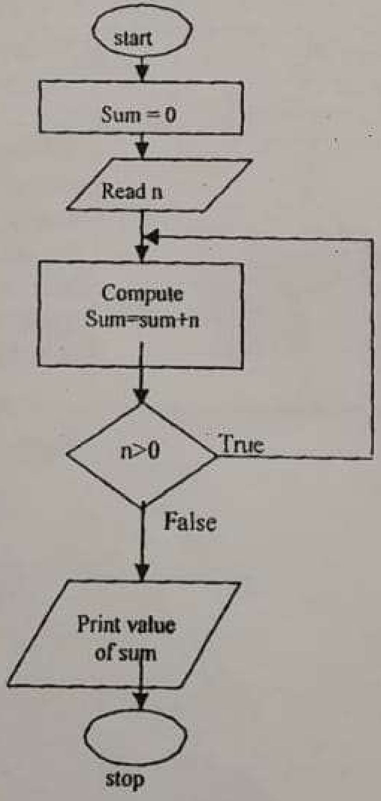
2.	**do-while statement**
	do
	{
	statement1;
	statement2;
	.
	.
	} while (condition);
		Loop Body
		True
		Test Condition
		False
			/*initialize sum*/
			sum =0;
			do
			{/* Get a number */
			printf(" give number"};
			scanf("%d",&n);
			/* add number to sum*/
			sum=sum+n;
			} while (n>0);
			printf ("sum is %d", sum);

3.	**for statement**
	
	for(expr1; expr2; expr3)
	{
	statement 1
	.
	.
	}
	expr1 = initialization expression
	expr2 = loop condition
	expr3 = alteration expression which alters the loop variable
			/* display first 10 multiples of 2 */
			for(i=1; i <= 10; i++)
			{
			printf ("2 X %d = %d\n", i, 2*i);
			}

Note: Usually the for loop is used when the statements have to executed for a fixed number of times. The while loop is used when the statements have to be executed as long as some condition is true and the do-while loop is used when we want to execute statements atleast once (example: menu driven programs)

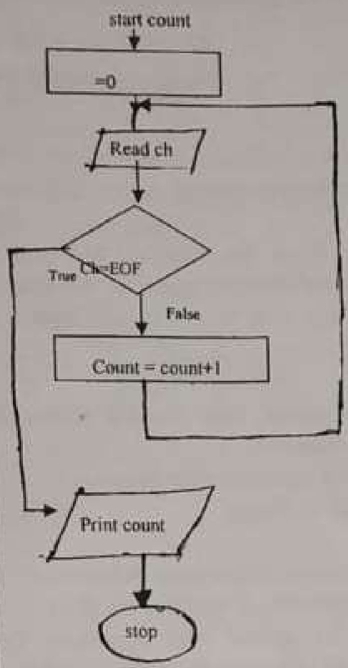
3. Sample program- to print sum of1+2+3+…..n.

Step 1: Writing the Algorithm	**Step 2: Draw the flowchart**	**Step 3: Writing Program**

1. Start
2. Initialize sum to 0.
3. Accept n.
4. Compute sum=sum+n
5. Decrement n by 1
6. if n >0
go to step 4
7. Display value of sum.
8. Stop
	 start
	Sum = 0
	Read n
	Compute Sum=sum+n
	n>0
	True
	False
	Print value of sum
	stop
		/* Program to calculate sum of numbers */
		
		#include <stdio.h>
		main( )
		{ /* variable declarations */
		int sum = 0, n;
		printf("enter the value of n : ");
		scanf("%d",&n);
		while (n>0)
		{
		sum = sum + n;
		n--;
		}
		printf("\n The sum of numbers is %d", sum);
		}

4. Sample program- To read characters till EOF (Ctrl+Z) and count the total number of characters entered.

Step 1 : Writing the Algorithm	**Step 2 : Draw the flowchart**	**Step 3 : Writing Program**

1. Start
2. Initialize count to0.
3. Accept ch.
4. If ch !=EOF
Count = count +1
Else
Go to step 6
5. Go to step3
7. Display value of sum.
8. Stop
	 start count
	=0
	Read ch
	True
	Ch=EOF
	False
	Count = count+1
	Print count
	stop
		/* Program to count number of characters
		*/
		
		#include <stdio.h>
		main( )
		{
		  char ch;
		  int count=0; while((ch=getchar())!=EOF)
		          count++;
		
		  printf("Total characters = %d", count);
		}

Set A . Apply all the three program development steps for the following examples.

1. Write a C program to accept n numbers from user and find sum of odds and evens
2. Accept an integer and find the sum of it's first and last digit.
3. Accept 2 numbers m and n. Display all those numbers between m and n which are divisible by 3 but not divisible by 7.
4. Accept 2 numbers. Find their GCD and display it.
5. Write a program to accept two integers x and n and compute x^n
6. Accept n numbers and count number of positive, negative and zero values.
7. Write a program to accept an integer and count the number of digits in the number.
8. Write a program to accept a character, an integer n and display the next n characters.

Signature of instructor □ Date / /

Set B. Apply all the three program development steps for the following examples.

1. Write a program to display the first n Fibonacci numbers. (1 1 2 3 5 ……)
2. Write a program to accept an integer and check if it is prime or not.

3. Write a program to accept an integer and reverse the number.
   Example: Input: 546, Output 645.
4. Write a program, which accepts a number n and display each digit in words.
   Example: 6702 Output=Six-Seven-Zero-Two
   (Hint: Reverse the number and use a switch statement)

Signature of instructor □ Date □ / /

Set C. Write C programs to solve the following problems

1. Write a program to accept characters from the user till the user enters * and count the number of characters, words and lines entered by the user. (Hint: Use a flag to count words. Consider delimiters like \n \t , ; . and space for counting words)

2. Write a program which accepts a number and checks if the number is a palindrome (Hint number = reverse of number)
   Example: number = 3472 Output: It is no t a palindrome number.
   = 262, Output: It is a palindrome

3. A train leaves station A at 4.00 am and travels at 80kmph. After every 30 minutes, it reaches a station where it halts for 10 minutes. It reaches its final destination B at 1.00 p.m. Display a table showing its arrival and departure time at every intermediate station. Also calculate the total distance between A and B.

Signature of Instructor □ Date □ / /

Assignment Evaluation

0: Not done □	2: Late Complete □	4: Complete □
1: Incomplete □	3: Needs improvement □	5: Well Done □

Signature