USN | | | | | | | | | | |

Internal Assessment Test 1 – Sept. 2017

Sub:	Programming in C & Data Structure					Sub Code:	15PCD13	Branch:	Chemistry Cycle
Date:	21-09-2017	Duration:	90 min's	Max Marks:	50	Sem / Sec:	I		OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1 (a)	What is a variable in C? Describe rules on naming variables. List FOUR valid and FOUR invalid variables with suitable reasons.	[2+2+2]	CO1	L2
(b)	Explain types of constants.	[04]	CO1	L1
2 (a)	Explain all operators used in C with examples.	[10]	CO1	L2
3 (a)	With Syntax and example explain printf and scanf functions.	[3+3]	CO1	L1
(b)	Evaluate the expression: i)3+2%10&&2 ii) 2|1-!10*++5	[2+2]	CO1	L3
4 (a)	Write a C program to calculate area and circumference of Circle.	[05]	CO2	L3
(b)	Write a program to swap two numbers and display numbers before swapping and after swapping.	[05]	CO2	L3
5 (a)	Explain if, if..else, nested if statements with syntax and suitable example.	[2+2+2]	CO2	L2
(b)	Explain break and continue statement with examples.	[2+2]	CO2	L2

USN | | | | | | | | | | |

Internal Assessment Test 1 – Sept. 2017

Sub:	Programming in C & Data Structure					Sub Code:	15PCD13	Branch:	Chemistry Cycle
Date:	21-09-2017	Duration:	90 min's	Max Marks:	50	Sem / Sec:	I		OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1 (a)	What is a variable in C? Describe rules on naming variables. List FOUR valid and FOUR invalid variables with suitable reasons.	[2+2+2]	CO1	L2
(b)	Explain types of constants.	[04]	CO1	L1
2 (a)	Explain all operators used in C with examples.	[10]	CO1	L2
3 (a)	With Syntax and example explain printf and scanf functions.	[3+3]	CO1	L1
(b)	Evaluate the expression: i)3+2%10&&2 ii) 2|1-!10*++5	[2+2]	CO1	L3
4 (a)	Write a C program to calculate area and circumference of Circle.	[05]	CO2	L3
(b)	Write a program to swap two numbers and display numbers before swapping and after swapping.	[05]	CO2	L3
5 (a)	Explain if, if..else, nested if statements with syntax and suitable example.	[2+2+2]	CO2	L2
(b)	Explain break and continue statement with examples.	[2+2]	CO2	L2

6	What is switch in C? Explain with syntax. Write a C program to accept an arithmetic operator (+,-,*,/,%) and two integers as input and display result of the operation as output. [Example: Input: + 2 3 Output: 5] .	[10]	CO2	L3
7	Write a C program to find roots of a quadratic equation of the form ax2+bx+c=0, where a, b and c are coefficients. Explain the output for all possible cases.	[10]	CO2	L3
8 (a)	With example explain the difference between while loop and do-while loop.	[6]	CO2	L3
(b)	Write a C program to check given year is leap year or not.	[4]	CO2	L3

6	What is switch in C? Explain with syntax. Write a C program to accept an arithmetic operator (+,-,*,/,%) and two integers as input and display result of the operation as output. [Example: Input: + 2 3 Output: 5] .	[10]	CO2	L3
7	Write a C program to find roots of a quadratic equation of the form ax2+bx+c=0, where a, b and c are coefficients. Explain the output for all possible cases.	[10]	CO2	L3
8 (a)	With example explain the difference between while loop and do-while loop.	[6]	CO2	L3
(b)	Write a C program to check given year is leap year or not.	[4]	CO2	L3

SCHEME

Question #	Description	Marks Distribution		Max Marks
1 a	variable in C rules on naming variables FOUR valid and FOUR invalid variables	2M 2M 2M	6M	6M
1 b	types of constants	4M	4M	4M
2	Explain all operators used in C with examples	10M	10M	10M
3 a	Syntax and example explain printf and scanf functions	3M 3M	6M	6M
3 b	i)3+2%10&&2 ii) 2|1-!10*++5	2M 2M	4M	4M
4 a	program to calculate area and circumference of Circle	5M	5M	5M
4 b	program to swap two numbers and display numbers before swapping and after swapping	4 M 1 M	5M	5M
5 a	if, if..else, nested if statements with syntax and suitable example.	2M 2M 2M	6M	6M
5 b	break and continue statement with examples	2M 2M	4M	4M
6	Switch statement with syntax Program simple calculator	4M 6M	10M	10M
7	program to find roots of a quadratic equation of the form ax2+bx+c=0, where a, b and c are coefficients	10M	10M	10M

8 a	difference between while loop and do-while loop	2M each	6M	6M
8 b	program to check given year is leap year or not.	4M	4M	4M

SOLUTION:

1 a.

- A variable is a data name that may be used to store a data value.
- Variable names may consist of letters, digits, and the underscore(_) character, subject to the rules given below:
  1. The variables must always begin with a letter. Some systems permit underscore as the first character.

  2. ANSI standard recognizes a length of 31 characters. However, the length should not be normally more than eight characters.

  3. Uppercase and lowercase are significant. That is ,the variable Rate is not the same as rate or TOTAL.

  4. The variable name should not be a keyword.

  5. White space is not allowed.
- Variables may take different values at different times during execution, constants remain unchanged during the execution.
- ```
  #include<stdio.h>
  int main()
  {
  inta,b,c=10,res; // a,b are varaiables& c is a constant with the value 10
  printf("\n Enter the values of a & b:");
  scanf("%d%d",&a,&b);
      res=a+b+c;
  printf("The sum is %d",res);
      return 0;
  }
  ```
- Valid variable examples abc, ab8gh, ab_cd, _abc
- Invalid variable examples 9ab, ab cd, ab**, _ab_cv7

…….
  ```

Q1 b.

Constants are expressions with a fixed value .
Different Types of C Constants :

Constant	**Type of Value Stored**
Integer Constant	Constant which stores integer value
Floating Constant	Constant which stores float value
Character Constant	Constant which stores character value
String Constant	Constant which stores string value

Declaration	**Explanation**
const int a = 1;	read as "a is an integer which is constant"
int const a = 1;	read as "a is a constant integer"

…….

Q 2.
Unary operators : +,-,++ --, !, &,
Binary Operators:
Supported Arithmetic Operators : 5 arithmetic operators are shown in the following table. Arithmetic operators are used to perform arithmetic operations in c programming.

Operator	**Meaning**	**Example**
+	Addition Operator	10 + 20 = 30
-	Subtraction Operator	20 – 10 = 10
*	Multiplication Operator	20 * 10 = 200
/	Division Operator	20 / 10 = 2
%	Modulo Operator	20 % 6 = 2

Logical Operator in C (results into 0 or 1)

Operator	**Name of the Operator**
&&	And Operator
||	Or Operator
!	Not Operator

Bitwise Operator in C
& And Operator
| **or operator**
^ **Ex-or**
<< left shift and right shift
>>

Shorthand Assignment operator
+=, *=, &= etc

Relational Operator (results into 0 or 1)
< <= > >= <> !=

Conditional Operators [?:] : Ternary Operator Statement in C
1. They are also called as Ternary Operator .
2. They also called as ?: operator
3. Ternary Operators takes on 3 Arguments

Syntax :
expression 1 ? expression 2 : expression 3
where
 expression1 is **Condition**

 expression2 is Statement Followed if **Condition is True**
 expression2 is Statement Followed if **Condition is False**

…….

Q3 a

the printf Function
The printf function is used to output any combination of numerical values, single characters and strings. The general form of the printf function is:
printf(control string, arg1, arg2, … , argN)
The **control string** consists of one or more of the following items:
Characters that are simply printed as they are; ie, messages or prompts
Conversion Specifications — the field width and the format specifier
Escape Sequences — used to print certain non-printing characters like newline, tab and bell alert
The control string indicates the type of arguments that follow. The arguments **arg1, arg2, … , argN** are the variables whose values are formatted and printed according to the specifications of the control string. The arguments must match in number, order and type with the format specifications.
Example:

```
#include<stdio.h>
int main()
{
int marks=412;
floatpercent=85.4;
printf("n Marks= %d,n Percentage=%f ",marks, percent);
return 0;
}
```

Output:
Marks= 412
Percentage= 85.4
In the above program, the escape sequence n has been used to print the output on multiple lines. The format specifiers %d and %f are used to indicate that the arguments marks and percentare integer type and floating type arguments respectively. The arguments to a printf function represents the **actual values** and not the addresses as was the case in scanf.

…….

Q3 b a) 1 b) 3 (solve and show)

…….

Q 4 a

```
#include<stdio.h>
#include<math.h>
int main()
{
int i;
    float degree, x;
// Input the value of x in degree.
printf ( "\nEnter the value of x in degree: " );
```

```
scanf ( "%f", &degree );
// Compute the value of x in radians.
x = (3.1412/180.0) * degree;
printf("\n The value of angle in radians is %f",x);
   return 0;
}
```

b.

```
int main()
{
int a=10,b=20,temp;
printf("\n Before swapping a=%d b=%d",a,b);
temp=a;
a=b;
b=temp;


printf("\n After swapping a=%d b=%d",a,b);
   return 0;
}
```

…….

Q5 a

If statement
Syntax :
if(expression)
statement1;
Explanation :
- Expression is Boolean Expression
- It may have true or false value

Meaning of If Statement :
- It Checks whether the given Expression is Boolean or not !!
- If Expression is True Then it executes the statement otherwise jumps to next_instruction

Sample Program Code :

```
void main()
{
int a=5,b=6,c;
c = a + b ;
if (c==11)
printf("Execute me 1");
printf("Execute me 2");
}
```

Output :
Execute me 1

If..else
If-Else Statement :

```
if(conditional)
{
//True code
}
else
{
//False code
}
```

Note :

Consider Following Example -

```
int num = 20;
if(num == 20)
{
printf("True Block");
}
else
{
printf("False Block");
}
```

1. If part Executed if Condition Statement is True.

```
if(num == 20)
{
printf("True Block");
}
```

True Block will be executed if condition is True.

2. Else Part executed if Condition Statement is False.

```
else
{
printf("False Block");
}
```

3. More than One Conditions can be Written inside If statement.

```
int num1 = 20;
int num2 = 40;
if(num1 == 20 && num2 == 40)
{
printf("True Block");
}
```

4. Opening and Closing Braces are required only when “Code” after if statement occupies multiple lines.
5. Code will be executed if condition statement is True.
6. Non-Zero Number Inside if means **“TRUE Condition”**

Nested if

If with in if: equivalent to logical && operation

```
if(condition)
{
        if(condition)
        {
        ….
        ….
        }
}
```

…….

Q 5b

Break and continue

There are two statements built in C programming, break; and continue; to alter the normal flow of a program. Loops perform a set of repetitive task until text expression becomes false but it is sometimes desirable to skip some statement/s inside loop or terminate the loop immediately without checking the test expression. In such cases, break and continue statements are used. The break; statement is also used in switch statement to exit switch statement.

break Statement

In C programming, break is used in terminating the loop immediately after it is encountered. The break statement is used with conditional if statement.

Syntax of break statement

break;

The break statement can be used in terminating all three loops for, while and do...while loops.

The figure below explains the working of break statement in all three type of loops.

Example of break statement

Write a C program to find average of maximum of *n* positive numbers entered by user. But, if the input is negative, display the average(excluding the average of negative input) and end the program.

```
/* C program to demonstrate the working of break statement by terminating a loop, if user inputs negative number*/
# include <stdio.h>
int main(){
float num,average,sum;
int i,n;
printf("Maximum no. of inputs\n");
scanf("%d",&n);
for(i=1;i<=n;++i){
printf("Enter n%d: ",i);
scanf("%f",&num);
if(num<0.0)
break; //for loop breaks if num<0.0
sum=sum+num;
}
average=sum/(i-1);
printf("Average=%.2f",average);
return 0;
}
```

Output

```
Maximum no. of inputs
4
Enter n1: 1.5
Enter n2: 12.5
Enter n3: 7.2
Enter n4: -1
Average=7.07
```

In this program, when the user inputs number less than zero, the loop is terminated using break statement with executing the statement below it i.e., without executing sum=sum+num.

In C, break statements are also used in switch...case statement.

continue Statement

It is sometimes desirable to skip some statements inside the loop. In such cases, continue statements are used.

Syntax of continue Statement

continue;

Just like break, continue is also used with conditional if statement.

For better understanding of how continue statements works in C programming. Analyze the figure below which bypasses some code/s inside loops using continue statement.

Example of continue statement

Write a C program to find the product of 4 integers entered by a user. If user enters 0 skip it.

```
//program to demonstrate the working of continue statement in C programming
# include <stdio.h>
int main(){
int i,num,product;
for(i=1,product=1;i<=4;++i){
printf("Enter num%d:",i);
scanf("%d",&num);
if(num==0)
continue; / *In this program, when num equals to zero, it skips the statement product*=num and continue the loop. */
product*=num;
}
printf("product=%d",product);
return 0;
}
```

Output

```
Enter num1:3
Enter num2:0
```

Enter num3:-5
Enter num4:2
product=-30

…….

Q 6

Switch Case :
switch(expression)
{
case value1 :
body1
break;
case value2 :
body2
break;
case value3 :
body3
break;
default :
default-body
break;
}
next-statement;
Flow Diagram :
*Steps are Shown in Circles.
How it works ?
 Switch case checks the value of expression/variable against the list of case values and when the match is found , **the block of statement associated with that case is executed**
 Expression should be Integer **Expression / Character**
 Break statement takes control out of the case.
 Break Statement is **Optional**.

```
# include <stdio.h>
int main() {
char o;
float num1,num2;
printf("Select an operator either + or - or * or / \n");
scanf("%c",&o);
printf("Enter two operands: ");
scanf("%f%f",&num1,&num2);
switch(o) {
case '+':
printf("%.1f + %.1f = %.1f",num1, num2, num1+num2);
break;
case '-':
printf("%.1f - %.1f = %.1f",num1, num2, num1-num2);
break;
case '*':
printf("%.1f * %.1f = %.1f",num1, num2, num1*num2);
break;
case '/':
printf("%.1f / %.1f = %.1f",num1, num2, num1/num2);
break;
default:
/* If operator is other than +, -, * or /, error message is shown */
printf("Error! operator is not correct");
break;
}
return 0;
}
```

Output

```
Enter operator either + or - or * or /
*
Enter two operands: 2.3
4.5
2.3 * 4.5 = 10.3
```

…….

Q 7

```
#include<stdio.h>
#include<math.h>
main()
{
float a,b,c,root1,root2,r_part,i_part,disc;
system("clear");
printf("\nEnter the 3 non-zero coefficients\n");
/* Read three non-zero coefficients such as a,b,c */
scanf("%f%f%f",&a,&b,&c);
disc=(b*b)-(4*a*c);
if(disc==0)
{
printf("\nRoots are equal\n");
root1=-b/(2*a);
root2=root1;
printf("\nroot1=root2=%f\n",root1);
}
else if(disc>0)
{
printf("\nRoots are real and distinct\n");
root1=(-b+sqrt(disc))/(2*a);
root2=(-b-sqrt(disc))/(2*a);
printf("\nroot1 = %f \t root2 = %f\n",root1,root2);
}
else
{
printf("\nRoots are complex\n");
r_part=-b/(2*a);
i_part=(sqrt(-disc))/(2*a);
printf("\nroot1 = %f+i%f \t root2 = %f-i%f\n",r_part,i_part,r_part,i_part);
}
```

…….

8 a.

Syntax of while loop

```
while (test expression) {
statement/s to be executed.
}
```

The while loop checks whether the test expression is true or not. If it is true, code/s inside the body of while loop is executed,that is, code/s inside the braces { } are executed. Then again the test expression is checked whether test expression is true or not. This process continues until the test expression becomes false.

Example of while loop

Write a C program to find the factorial of a number, where the number is entered by user. (Hints: factorial of n = 1*2*3*...*n

```
/*C program to demonstrate the working of while loop*/
#include <stdio.h>
int main(){
int number,factorial;
printf("Enter a number.\n");
```

```
scanf("%d",&number);
factorial=1;
while (number>0){ /* while loop continues util test condition number>0 is true */
factorial=factorial*number;
--number;
}
printf("Factorial=%d",factorial);
return 0;
}
```

Output

```
Enter a number.
5
Factorial=120
```

do...while loop

In C, do...while loop is very similar to while loop. Only difference between these two loops is that, in while loops, test expression is checked at first but, in do...while loop code is executed at first then the condition is checked. So, the code are executed at least once in do...while loops.

Syntax of do...while loops

```
do {
some code/s;
}
while (test expression);
```

At first codes inside body of do is executed. Then, the test expression is checked. If it is true, code/s inside body of do are executed again and the process continues until test expression becomes false(zero).

Notice, there is semicolon in the end of while (); in do...while loop.

Example of do...while loop

Write a C program to add all the numbers entered by a user until user enters 0.

```
/*C program to demonstrate the working of do...while statement*/
#include <stdio.h>
int main(){
int sum=0,num;
do /* Codes inside the body of do...while loops are at least executed once. */
{
printf("Enter a number\n");
scanf("%d",&num);
sum+=num;
}
while(num!=0);
printf("sum=%d",sum);
return 0;
}
```

Output

```
Enter a number
3
Enter a number
-2
Enter a number
0
sum=1
```

…….

8b.

```
#include <stdio.h>
int main()
{
    int year;
    printf("Enter a year: ");
    scanf("%d",&year);
    if(year%4 == 0)
```

```
    {
       if( year%100 == 0) /* Checking for a century year */
       {
          if ( year%400 == 0)
            printf("\n%d is a leap year.\n", year);
          else
            printf("\n%d is not a leap year.\n", year);
       }
       else
         printf("\n%d is a leap year.\n", year );
    }
    else
      printf("\n%d is not a leap year.\n", year);
    return 0;
}
```

…….