

USN

1 P E



PESIT Bangalore South Campus

Hosur Road (1km before Electronic City), Bengaluru -560 100

Department of Basic Science and Humanities

SOLUTION OF CONTINUOUS INTERNAL EVALUATION TEST -1

Date : 27 - 02 - 2018

Marks:60

Subject & Code : **Programming in C and Data Structures- 17PCD23**Section: **F, G, H, I, J**Name of faculty : Dr. J Surya Prasad/Mr. Naushad Basha Saudagar /
Ms. Kakoli Bora / Ms. Monika

Time : 11:30 am – 1:00 pm

Note: Answer FIVE full questions choosing any ONE full question from each part.

PART 1

1	A	Define an algorithm. Write an algorithm to find the area and perimeter of a rectangle.	6
	Ans:	<p>An Algorithm is a step by step procedure to solve a given problem in finite number of steps by</p> <ul style="list-style-type: none"> <input type="checkbox"/> Accepting a set of inputs and <input type="checkbox"/> Producing the desired output for the given problem. <p>Step 1: [start] Step 2: [input the value of length and breadth] read l and b Step 3: [calculate the area of rectangle] Area= l*b Step 4: [calculate the perimeter of rectangle] perimeter= 2*(l+b) Step 5: [display the calculated area and perimeter] write Area and perimeter Step 6: [finished] Stop</p>	6
	B	<p>Draw the flowchart and write a C program to calculate factorial of a number.</p> <pre> #include<stdio.h> void main() { int n,f=1; printf("enter the number"); scanf("%d",&n); while(n!=0) { f=f*n; n--; } printf("Factorial = %d",f); } </pre> <pre> graph TD Start([Start]) --> Input[/Input n/] Input --> Decision{Is (n!=0)} Decision -- true --> Process[f=f*n; n=n-1] Process --> Decision Decision -- false --> Print[/Print '!'/] Print --> Stop([Stop]) </pre>	6

USN	1	P	E							
	PESIT Bangalore South Campus Hosur Road (1km before Electronic City), Bengaluru -560 100 Department of Basic Science and Humanities									

2	A	What is a Variable? Explain the rules for constructing variables in C language. Give examples for valid and invalid variables.	6
	Ans:	<p>A variable is an identifier whose value can be changed during execution of the program.</p> <p>Σ The memory location name where the data can be stored.</p> <p>Rules for defining a variable:</p> <ol style="list-style-type: none"> 1) The first character in the variable name must be an alphabet or underscore. 2) No commas or blanks are allowed within a variable name. 3) No special symbol other than an underscore can be used in a variable name. 4) A variable name is any combination of alphabets, digits or underscores. <p>Valid Variables: a_b, g1, rem2</p> <p>Invalid Variables: 1a, h\$</p>	
	B	Temperature of a city in Fahrenheit degrees is input through the keyboard. Write a program to convert the temperature into Centigrade degrees.	6
	Ans:	<pre>#include<stdio.h> void main() { float f,c; printf("enter the temp in Fahrenheit"); scanf("%f", &f); c = ((float)5/9)*(f-32); printf("Temperature in centigrade degrees = %f", c); }</pre>	
PART 2			
3	A	Explain the primary data types and their variants. Also mention the format specifiers for the same.	6
	Ans:	<p>Data Types: The data type defines the types of data stored in a memory location.</p> <p>Character Integer Float Void</p> <p>Character Data type: Keyword for character is 'char'. It takes 1 byte in memory.</p> <p>Chars, signed and unsigned: Signed: When the variable is having signed like positive or negative. Unsigned: When the variable value is having no signed.</p> <p>A signed char is same as an ordinary char and has a range from -128 to 127; whereas, an unsigned char has a range from 0 to 255.</p> <p>Integer Data Type: Keyword for integer is 'int'.</p> <p>Integer, long and short: Integer can also be classified as short int and long int. The intension of providing these variations is to provide integers with different ranges.</p> <p>Example: short int I; or short I; long int j; or long j;</p> <p>Integers, signed and unsigned: When the value stored in a given integer variable having positive or negative signed that comes under signed integer.</p> <p>Example: signed int I; When the variable is being used to only count things having no signed then it comes under unsigned integer.</p> <p>Example: unsigned int j;</p> <p>Floats and Doubles: A float occupies 4 bytes in memory. If this is insufficient, then C offers a double data type that occupies 8 bytes in memory. If this is insufficient, then C offers a long double data type that occupies 10 bytes in memory.</p>	
	B	Write a C program that computes the size of int, float, double and char variables.	6
	Ans:	#include<stdio.h>	

USN	1	P	E							
	PESIT Bangalore South Campus Hosur Road (1km before Electronic City), Bengaluru -560 100 Department of Basic Science and Humanities									

		<pre>void main() { int a; char b; float c; printf(“ Size of integer=%d, size of character=%d, size of float=%d”,sizeof(a),sizeof(b),sizeof(c)); }</pre>	
4	A	What do you mean by unary, binary and ternary operator explain briefly?	6
	Ans:	<p>1) Unary Operators: Unary operations operate on a single operand, therefore the number 5 when operated by unary – will have the value –5.</p> <p>a) Addition or Unary plus (+) . example: a = +5</p> <p>b) Subtraction or Unary Minus (-) . Example: a= -7</p> <p>c) Increment or Decrement operator (++ , --) . Example: If a=5 then ++a will be a=6, b=4 then b—will be b=3.</p> <p>d) Logical Not (!) . Example: A=1 then (! A) will be A=0</p> <p>e) Bitwise Complement (~) . Example: A=1001 then (~A) will be A=0110</p> <p>f) Address of (&). Example: printf(“%d”,&A) will print the address of A</p> <p>g) Pointer dereference (*). Example: printf(“%d”,*a) will print the value at that address.</p> <p>2) Binary operators: A binary operator is an operator that operates on two operands and manipulates them to return a result. Operators are represented by special characters or by keywords and provide an easy way to compare numerical values or character strings. Binary operators are presented in the form: Operand1 Operator Operand2.</p> <p>(a) Arithmetic Operators (+,-,*,/,%)</p> <p>(b) Relational Operators (<,<=,>,>=,==,!=)</p> <p>(c) Logical Operators (&&,)</p> <p>(d) Assignment Operators and shorthand assignment operators (=,+ =,- =,etc.)</p> <p>(e) Bitwise Operators (& , ,^ ,<< ,>>)</p> <p>3) Ternary Operators: a ternary operator is an operator that takes three arguments. The arguments and result can be of different types. Many programming languages that use C-like syntax feature a ternary operator, ?: , which defines a conditional expression. Exp1?Exp2:Exp3 If exp1 is true ans will be exp 2 otherwise ans will be exp3.</p>	
	B	Two numbers are input through the keyboard into two locations a and b. WAP to interchange the contents of a and b using third variable.	6
	Ans:	<pre>#include<stdio.h> void main() { int a, b, c; printf(“enter the values of a and b\n”); scanf(“%d%d”,&a,&b); printf(“Before swapping values are a=%d and b=%d\n”,a,b); c=a; a=b; b=c; printf(“After swapping values are a=%d and b=%d\n”,a,b); }</pre>	
PART 3			
5	A	Evaluate the expressions where a=2, b=8, c=4, d=10	6

USN

1

P

E



PESIT Bangalore South Campus

Hosur Road (1km before Electronic City), Bengaluru -560 100

Department of Basic Science and Humanities

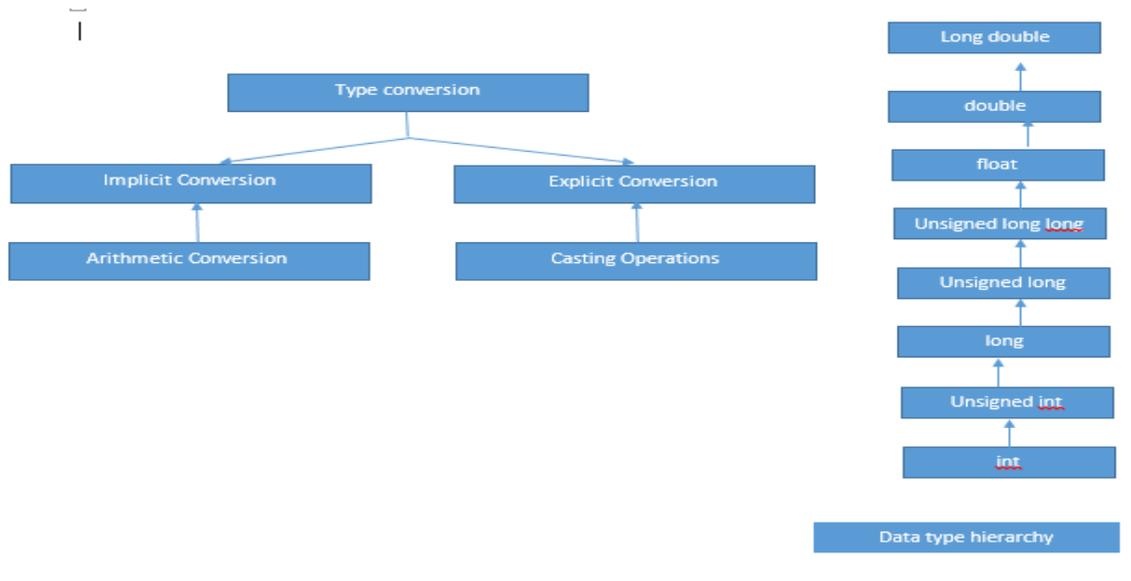
	(i) $a + b/c*d-c/a$ (ii) $(b/a)\%c$ (iii) $a++ + b-- + d++$	(iv) $a+=b*=c-=5$ (v) $2*((a\%5)*(4+(b-3)/c+2)))$ (vi) $100\%20<=20-5+100\%10-20==5>=1!=20$	
Ans:	(i) 20 (ii) 0 (iii) 20	(iv) -6 (v) 28 (vi) 1	
B	What do you do you understand by Operator Precedence and Associativity.		6
Ans:	<p>Operator Precedence and Associativity</p> <p>Each operator in c has a precedence associated with it. This precedence is used to determine how an expression involving more than one operator is evaluated. There are distinct levels of precedence and an operator may belong to one of these levels. The operators at the higher level of precedence are evaluated first. The operators of the same precedence are evaluated either from “left to right” or from “right to left” depending on the level. This is known as associativity property of an operator.</p> <p>Rules for evaluation of expression</p> <ul style="list-style-type: none"> • First parenthesized sub expression left to right are evaluated. • If parentheses are nested, the evaluation begins with the innermost sub expression. • The precedence rule is applied in determining the order of application of operators in evaluating sub expressions. • The associability rule is applied when two or more operators of the same precedence level appear in the sub expression. • Arithmetic expressions are evaluated from left to right using the rules of precedence. • When Parenthesis is used, the expressions within parenthesis assume highest priority. <p>Example:</p> $10 - 3 \% 8 + 6 / 4$ $10 - 3 + 6 / 4$ $10 - 3 + 1$ $7 + 1$ 8		
6 A	What do you mean type Conversion? Explain the two types of conversion with examples.		6
Ans:	<p>TYPE CONVERSION</p> <ul style="list-style-type: none"> • Type conversion is used to convert data of one type to data of another type. • Type conversion is of 2 types as shown in below figure: 		



PESIT Bangalore South Campus

Hosur Road (1km before Electronic City), Bengaluru -560 100

Department of Basic Science and Humanities



IMPLICIT CONVERSION

- If a compiler converts one type of data into another type of data automatically, it is known as implicit conversions.
- There is no data loss in implicit conversion.
- The conversion always takes place from lower rank to higher rank.
For ex, int to float as shown in the above datatype hierarchy.
- For ex:


```
int a = 22, b=11;
float c = a; //c becomes 21.000000
float d=b/c=11/22.000000=11.000000/22.000000=0.500000
```
- If one operand type is same as other operand type, no conversion takes place and type of result remains same as the operands i.e. int+int=int
float+float=float
- Conversion rules are as follows:
 - If either operand is long double, convert the other to long double.
 - Otherwise, if either operand is double, convert the other to double.
 - Otherwise, if either operand is float, convert the other to float.
 - Otherwise, convert char and short to int.
 - Then, if either operand is long, convert the other to long.
- Example: Program to illustrate implicit conversion.

```
#include<stdio.h> void
main()
{int a = 22, b=11; float d ;
  d=b/c;
  printf("d Value is : %f ", d );
}
```

Output:
d Value is : 0.500000

EXPLICIT CONVERSION

- When the data of one type is converted explicitly to another type with the help of some pre-defined functions, it is called as explicit conversion.
- There may be data loss in this process because the conversion is forceful.
- The syntax is shown below:



PESIT Bangalore South Campus

Hosur Road (1km before Electronic City), Bengaluru -560 100

Department of Basic Science and Humanities

```
data_type1 v1;
data_type2 v2= (data_type2) v1;
    where v1 can be expression or variable
```

• For ex:

```
float b=11.000000; int c =
22;
float d=b/(float)c=11.000000/22.000000=0.500000
```

• Example: Program to illustrate explicit conversion.

```
#include<stdio.h>
void main()
{ float b=11.000000; int c = 22;
float d;
d=b/(float)c;
printf("d Value is : %f ", d); }
```

Output: d Value is : 0.500000

B Design and develop a C program to read a year as an input and find whether it is leap year or not. Also consider end of the centuries. **6**

Ans:

```
#include<stdio.h>
#include<math.h>
void main()
{
    int year;

    printf("Enter the Year");
    scanf("%d",&year);
    if(year % 4==0 && year % 100!=0||year%400==0)
        printf("\n\n%d is leap year\n",year);
    else
        printf("\n\n%d is not leap year\n",year);
}
```

PART 4

7 A Design and develop a C program to reverse of an integer number NUM and Check whether it is PALINDROME or NOT. **6**

Ans:

```
#include<stdio.h>
void main()
{
    int m,NUM,rev=0,rem;

    printf("Enter the number\n");
    scanf("%d",&NUM);
    m=NUM;
    while(NUM!=0)
    { rem=NUM%10;
rev=rev*10+rem;
NUM=NUM/10;
}
if(rev==m)
```



PESIT Bangalore South Campus

Hosur Road (1km before Electronic City), Bengaluru -560 100

Department of Basic Science and Humanities

```

printf("%d is palindrome\n",m);
else
printf("%d is not palindrome\n",m);
printf("Reverse of the number is %d\n",rev);
}

```

B Write a C program to calculate area of circle, rectangle and triangle using switch statement.

6

Ans:

```

#include <stdio.h>
void main ()
{
int choice, r ,l, w, b, h;
float area;
printf("Input 1 for area of circle\n");
printf("Input 2 for area of rectangle\n");
printf("Input 3 for area of triangle\n");
printf("Input your choice : ");
scanf("%d",&choice);
switch(choice)
{
case 1:
printf("Input radius of the circle : ");
scanf("%d",&r);
area=3.14*r*r;
break;
case 2:
printf("Input length and width of the rectangle : ");
scanf("%d%d",&l,&w);
area=l*w;
break;
case 3:
printf("Input the base and hight of the triangle :");
scanf("%d%d",&b,&h);
area=.5*b*h;
break;
}
printf("The area is : %f\n",area);
}

```

8 A List and explain all the conditional control statements used in C. Write a C program to find the Largest of three numbers.

12

Ans: Conditional statements in C programming language are

1. if statement
2. if-else statement
3. ternary statement or ternary operator
4. nested if-else statement
5. Else- if ladder



PESIT Bangalore South Campus

Hosur Road (1km before Electronic City), Bengaluru -560 100

Department of Basic Science and Humanities

6. switch statement

THE if STATEMENT

This is basically a “one-way” decision statement.

This is used when we have only one alternative.

The syntax is shown below:

```
if(expression)
{
statement1;
}
```

- Firstly, the expression is evaluated to true or false.

If the expression is evaluated to true, then statement1 is executed.

If the expression is evaluated to false, then statement1 is skipped.

THE if else STATEMENT

- This is basically a “two-way” decision statement.
- This is used when we must choose between two alternatives.
- The syntax is shown below:

```
if(expression)
{
statement1;
}
else
{
statement2;
}
```

- Firstly, the expression is evaluated to true or false.

If the expression is evaluated to true, then statement1 is executed.

If the expression is evaluated to false, then statement2 is executed.

THE nested if STATEMENT

- An if-else statement within another if-else statement is called nested if statement.
- This is used when an action has to be performed based on many decisions. Hence, it is called as multi-way decision statement.
- The syntax is shown below:

```
if(expr1)
{
if(expr2)
statement1
```



PESIT Bangalore South Campus

Hosur Road (1km before Electronic City), Bengaluru -560 100

Department of Basic Science and Humanities

```
else
```

```
    statement2
```

```
}
```

```
else
```

```
{
```

```
    if(expr3)
```

```
        statement3
```

```
    else
```

```
        statement4
```

```
}
```

- Here, firstly expr1 is evaluated to true or false.

If the expr1 is evaluated to true, then expr2 is evaluated to true or false. If the expr2 is evaluated to true, then statement1 is executed.

If the expr2 is evaluated to false, then statement2 is executed.

If the expr1 is evaluated to false, then expr3 is evaluated to true or false. If the expr3 is evaluated to true, then statement3 is executed.

If the expr3 is evaluated to false, then statement4 is executed.

THE else if LADDER STATEMENT

- This is basically a “multi-way” decision statement.
- This is used when we must choose among many alternatives.
- The syntax is shown below:

```
if(expression1)
```

```
    statement1; else
```

```
    if(expression2)
```

```
        statement2;
```

```
    else if(expression3)
```

```
        statement3
```

```
    else if(expression4)
```

```
        statement4
```

```
    else
```

```
        default statement5
```

- The expressions are evaluated in order (i.e. top to bottom).
- If an expression is evaluated to true, then
 - statement associated with the expression is executed &
 - control comes out of the entire else if ladder



PESIT Bangalore South Campus

Hosur Road (1km before Electronic City), Bengaluru -560 100

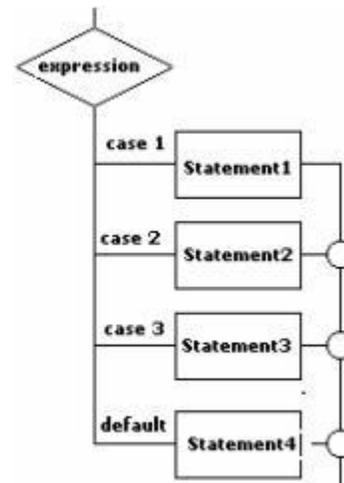
Department of Basic Science and Humanities

- For ex, if expression1 is evaluated to true, then statement1 is executed.

If all the expressions are evaluated to false, the last statement4(default case) is executed.
THE switch STATEMENT

- This is basically a “multi-way” decision statement.
- This is used when we must choose among many alternatives.
- The syntax & flow diagram is shown below:

```
switch(choice)
{
    case value1: statement1
                break;
    case value2: statement2
                break;
    case value3: statement3
                break;
    default: statement4;
}
```



- Here, choice can be either any integer value or a character.
- Based on this integer value, the control is transferred to a particular case-value where necessary statements are executed.
- During executing, if break statement is encountered, then the control comes out of the switch block.
- If the value of the choice does not match with any of the case values (i.e. value1, value2, value3) then control goes to default label.
- All case-values must be different.

Program:

```
: #include<stdio.h>
    void main() {
        int a,b,c; clrscr();
        printf("enter the numbers);
        scanf("%d%d%d",&a,&b,&c);
        if(a>b&&a>c)
            printf("A is largest");
        else if (b>a&&b>c)
            printf("B is largest");
        else
            printf("C is largest");
    }
```

USN	1	P	E							
	PESIT Bangalore South Campus Hosur Road (1km before Electronic City), Bengaluru -560 100 Department of Basic Science and Humanities									

PART 5			
9	A	Write a program to determine whether a given number is “odd” or “even” and print the Message ‘NUMBER IS EVEN’ OR ‘NUMBER IS ODD’, (A) Without using else option and (B) With else option.	12
	Ans:	<pre>A) #include <stdio.h> int main() { int number; printf("Enter an integer: "); scanf("%d", &number); (number % 2 == 0) ? printf("%d is even.", number) : printf("%d is odd.", number); return 0; } B) #include <stdio.h> int main() { int number; printf("Enter an integer: "); scanf("%d", &number); // True if the number is perfectly divisible by 2 if(number % 2 == 0) printf("%d is even.", number); else printf("%d is odd.", number); return 0; }</pre>	
10	A	Design and develop a flowchart that takes three coefficients (a, b, and c) of a Quadratic equation ($ax^2 + bx + c = 0$) as input and compute all possible roots. Implement a C program for the developed flowchart and execute the same to output the possible roots for a given set of coefficients with appropriate messages.	12
	Ans:	<pre>#include <stdio.h> #include <math.h> void main() { float a,b,c,D,r1,r2,m,s; printf("ENTER the coefficients of x2,x and constant\n");</pre>	



PESIT Bangalore South Campus

Hosur Road (1km before Electronic City), Bengaluru -560 100

Department of Basic Science and Humanities

```
scanf("%f%f%f",&a,&b,&c);
D=(b*b-4*a*c);
if ((a==0)||(b==0)
printf("Invalid inputs");
else if(D>0)
    {
        printf("the roots are real and distinct");
        r1=(-b+pow(D,0.5))/(2*a);
        r2=(-b-pow(D,0.5))/(2*a);
        printf("The roots are %f %f",r1,r2);
    }
else if(D==0)
    {
        printf("The roots are real and equal \n");
        r1=(-b/(2*a));
        r2=r1;
        printf("the roots are %f,%f",r1,r2);
    }
else
    {
        D=-D;
        printf("the roots are imaginary");
        m=(-b/(2*a));
        s=(pow(D,.5)/(2*a));
        printf("the roots are %f+i%f and %f-i%f",m,s,m,s);
```



PESIT Bangalore South Campus

Hosur Road (1km before Electronic City), Bengaluru -560 100

Department of Basic Science and Humanities

