ANNA UNIVERSITY SOLVED QUESTION PAPERS

B.E/B.Tech. Degree Examination, January 2014
GE6151 – Computer Programming
(Common to all branches)
(Regulation 2013)

Time : Three hours **Maximum : 100 Marks**

PART–A (10 × 2 = 20 marks)

1. List some important hardware and software technologies of the fifth generation computers.

- Magnetic drums were used for memory.
- Computers were difficult to program since they used machine language.
- Punch cards and paper tapes were used as inputs and output was displayed on printouts.

2. Write two characteristics of pseudo code.

- Since it is independent of any language, it can be used by most programmers.
- It is easy to develop a program from pseudo code than with a flowchart.
- It is easy to translate pseudo code into a programming language.
- Pseudo code is compact and does not tend to run over many pages.

3. What are the various types of C operators?

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Assignment Operator
- Increment & decrement Operator
- Conditional Operator
- Bitwise Operator
- Special Operator

4. Write a for loop statement to print numbers from 10 to 1.

```
#include<stdio.h>
main()
{
inti;
for(i=10;i<=1;i--)
{
```

```
printf("%d", i);
}
```

5. Define anarray.

An array is a derived data type. It is a collection of data elements of similar data type under a common name. Each element in the collection is identified by an index. The data elements are stored in contiguous memory location.

General syntax for an array:
datatype array_name[n]
Example, inta[10];

6. Name any two library functions used for string handling.

1. intstrlen(string_var)	to compute the length of the string,not counting null character
2. strcpy(dst_string,src_string)	to copy a source string into destination string
3. intstrcmp(str1,str2)	to compare str1 with str2
4. strcat(dst_string,src_string)	to append copy of the source string at the end of destination string

7. What is the need for functions?

- Functions are sub programs which perform a specific task.
- Functions support modularity in software engineering. That is, a complex problem is divided into easily manageable modules of functions.
- Similarly repetitive operations are specified within the function only once and it can be invoked in several places. So functions are used to reduce the length of the program.

8. What are the uses of pointers?

- Pointers save the memory space.
- Execution time with pointer is faster because data is manipulated with the address.
- The memory is accessed efficiently with the pointers.
- Dynamically, memory is allocated.
- Pointers are useful for representing two dimensional and multidimensional arrays.

9. Write any two preprocessor directives in C.

#include
#define
#ifdef
#ifndef
#pragma

10. Differentiate between structure and union.

S. No.	Structure	Union
1.	The keyword is struct.	The keyword is union.

2.	Memory allocation is done for all the data members in the structures.
	Example:
	struct student
	{
	introllno;
	char name[5];
	}s1;
	The memory allocation is 7 bytes.
		Memory allocation is done for the data member that requires maximum allocations.
		Example:
		union student
		{
		introllno;
		char name[5];
		}
		The memory allocation is 5 bytes.

3.	All the data members are available in the primary memory at any time of execution.	Only the last stored data element is available in the primary memory at any time of execution.
4.	Since memory is allocated for all the data members, no data is deleted in the primary memory.	Since memory is not allocated for all the data members, only one data is available and other data is deleted from the primary memory.

PART–B (5 × 16 = 80 marks)

11. (a) Write in detail about the evolution and the various generations of computers. (16)
Refer 1.4 & 1.5

Or

(b) Explain the basic computer organization using a neat diagram. (16)
Refer 2.1 & 2.2

12. (a) Write about the need and types of looping statements in C language and discuss with examples. (16)
Refer 11.1, 11.2,11.3 &11.4

Or

(b) Write about the need and types of branching statements in C language and discuss with examples. (16)
Refer 10.1 & 10.2

13. (a) (i) Write a C program to reverse a string. (8)
Refer Appendix–I, Ex. 48

(ii) Write a C program to print the Fibonacci series of a given number. (8)

```
#include<stdio.h>
```

```
int main()
{
int n, first = 0, second = 1, next, c;
printf("Enter the number of terms\n");
scanf("%d",&n);
printf("First %d terms of Fibonacci series are :-\n",n);
for ( c = 0 ; c < n ; c++ )
    {
if ( c <= 1 )
next = c;
else
    {
next = first + second;
first = second;
second = next;
    }
printf("%d\n",next);
    }
return 0;
}
```

Or

(b) Write a C program to print the sum of two matrices. (16)
Refer Appendix–I, Ex. 22 & 14.10

14. (a) Explain the following with suitable examples:

(i) Function declaration (8)
Refer 14.4

(ii) Call by reference,Call by value (8)
Refer 14.10 and 14.11

Or

(b) (i) Explain function with and without arguments with examples for each. (10)
Refer 14.6

(ii) What is recursion? Give an example. (6)
Refer 14.13.

15. (a) (i) What is storage class? List and explain with the help of an example. (8)
Refer 17.1.

(ii) Define and declare a structure to store date, which includes day, month and year. (8)

```
#include<stdio.h>
#include<conio.h>
```

```
struct date
{
int day;
int month;
int year;
};

void main()
{
struct date d1;
clrscr();
printf(“Enter  date(dd/mm/yyyy):”);
scanf(“%d%d%d”,&d1.day,&d1.month,&d1.year);
printf(“\nYour Date is :%d%d%d”,d1.day,d1.month,d1.year”);
getch();
}
```

Or

(b) Write a C program to create a mark sheet for students using structure. (16)
Refer 16.4

B.E/B.Tech. Degree Examination, May 2014
GE6151 – Computer Programming
(Common to all branches)
(Regulation 2013)

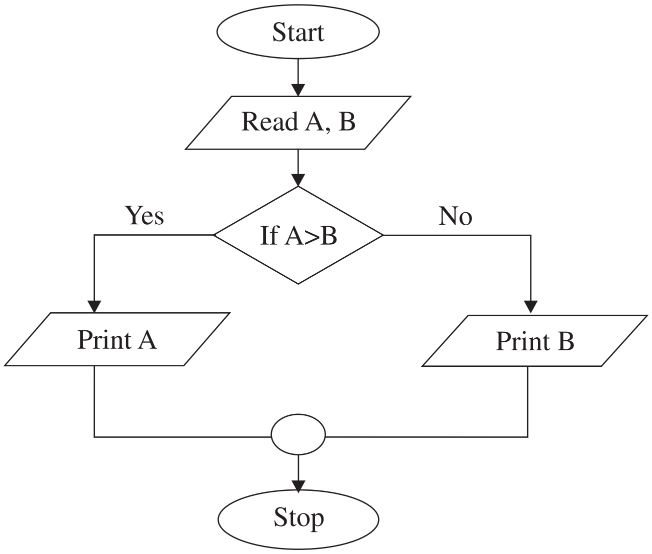
Time : Three hours **Maximum : 100 Marks**

PART–A (10 × 2 = 20 marks)

1. Convert the binary number 110 100 111 101 to octal.

$(110\ 100\ 111\ 101)_2 = (6475)_8$

2. Draw a flowchart to find the bigger of the two numbers.



3. List the different data types available in C.

C Data Types			
Primary	**User Defined**	**Derived**	**Empty**
Char int float double	typedef	arrays pointers structures union	void

4. Write a C program to find factorial of a given number using iteration.
Refer Appendix–I, Ex. 7

5. Write an example code to declare two dimensional arrays.

```
int  a[3][2], b[3][3];
```

Herea[3][2] means array 'a' contains 3 rows and 2 columns. So in total (3*2) = 6 data in the table. Similarly array 'b' contains 3 rows and 3 columns,in total 9 data in the table. Two dimensional arrays are used to handle matrix elements conveniently.

6. List any four string handling functions.

1.intstrlen(string_var)	to compute the length of the string,not counting null character
2.strcpy(dst_string,src_string)	to copy a source string into destination string
3.intstrcmp(str1,str2)	to compare str1 with str2
4.strcat(dst_string,src_string)	to append copy of the source string at the end of destination string

7. Define recursion.

When a function calls itself, then it is called recursive function or function recursion. Recursive function must include a conditional statement which forces to exit from the execution of body of the recursive function. Otherwise the function will never stop its execution.

8. What is the difference between call by value and callby reference?

Call by value It means a programmer sends some value coping from one function to another. At the time of function calling, a programmer can send a copy of variable of value.

Call by reference It means sending the address of variable to the called function, i.e.,a user can send the address of variable.

9. What is the purpose of Unions in C?

Memory allocation is done for the data member that requires maximum allocations. Hence less memory space is needed and so Union is used in C Programming.

10. What is the purpose of preprocessor directives?

The preprocessor examines the code before actual compilation of code begins and resolves all these directives before any code is actually generated by regular statements.

These lines are not program statements but directives for the preprocessor.

PART–B (5 × 16 = 80 marks)

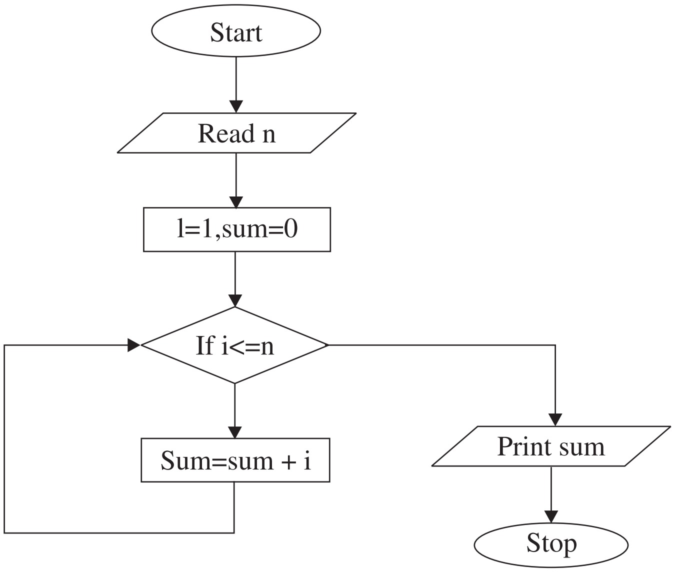
11. (a) (i) Discuss about generation of digital computers. (10)
Refer 1.6

(ii) Draw a flow chart to find the factorial of a number. (6)
Refer 5.2

Or

(b) (i) Explain the basic organization of a computer. (10)
Refer 2.1

(ii) Draw a flow chart to find the sum of first 100 natural numbers. (6)



12. (a) (i) Write a program to check whether a given number is prime or not. (8)

```
#include<stdio.h>
int main()
{
   int num,i,count=0;
   printf("Enter a number: ");
   scanf("%d",&num);
   for(i=2;i<=num/2;i++){
      if(num%i==0){
       count++;
          break;
      }
   }
  if(count==0 &&num!= 1)
      printf("%d is a prime number",num);
  else
```

```
        printf("%d is not a prime number",num);
      return 0;
    }
```

(ii) Write a C program to find the sum of the digits of an integer. (8)
Refer Appendix–I, Ex. 14

Or

(b) (i) Write a C program to find the roots of the quadratic equation. (8)
Refer Appendix–I, Ex. 50

(ii) Differentiate entry and exit checked conditional constructs with an example. (8)
Refer 11.1

13. (a) (i) Explain the concept of call by reference with suitable example. (8)
Refer 14.11

(ii) Write a C program to find the factorial of a number using recursion. (8)
Refer 14.13

Or

(b) (i) Write a C program to swap the content of two variables using pointers. (8)
Refer Appendix–I, Ex. 17

(ii) Explain the use of pointers in arrays with examples. (8)
Refer 10.10

14. (a) Write a C program to arrange the numbers in ascending order. (16)
Refer Appendix–I, Ex. 19

Or

(b) Write a C program to subtract two matrices and display the resultant matrix. (16)
Refer 12.14

15. (a) Explain the concept of storage classes with suitable example. (16)
Refer 17.1

Or

(b) Write a C program to store the employee's information using structure and search a particular employee using Employee Number. (16)

```
#include<stdio.h>
#include<conio.h>
structemp
{
 intempno;
 char ename[10];
 intesal;
} ;
main()
{
   structemp e[10];
   intn,i, eno, c =0;
   clrscr();
   printf(“\n How many records”);
   scanf(“%d”,&n);
   printf(“\n Enter the records”);
   for(i=0;i<n;i++)
   scanf(“%d%s%d”,&e[i].empno,e[i].ename,&e[i].esal);
    printf(“\n EnterEmployee No. to search:”);
   scanf(“%d”,&eno);
   for(i=0;i<n;i++)
            {
                  if(eno == e[i].empno)
                  {
                  printf(“\n %d\n %s\n %d”,e[i].empno,e[i].ename,e[i].esal);
                  c = c+1;
                  }
            }
if (c<=0)
printf( “ No Record Found”);
   getch();
}
```

B.E/B.Tech. Degree Examination, Nov/Dec 2014
GE6151 – Computer Programming
(Common to all branches)
(Regulation 2013)

Time : Three hours **Maximum : 100 Marks**

PART–A (10 × 2 = 20 marks)

1. Define a flowchart. Why is it required?
Flowchart is a pictorial representation of an algorithm in which the steps are drawn in the form of different shapes and the logical flow is indicated by arrows.

2. What is an algorithm?
Algorithm is defined as the finite sequence of explicit instructions that when provided with a set of input values processes an output and then terminated.

3. What is a variable? Illustrate with an example.
A variable is a data name that is used to store a data value. The variable may take different values at different times during execution.
Example: int a; // where ‘a’ is an integer variable.

4. Give an example for ternary operator?
Syntax: ? :
Example: 1<2 ? 1 : 2

5. Declare a float array of size 5 and assign 5 values to it.
float a[5] = {1.11,2.22,3.45,4.56,5.77,7.44};

6. Give an example for initialization of string array.
string name[ ] ={ “st.”,”joseph’s”,”institute”,”of”,”technology”);

7. What is a function?
A function is an independent program that performs a specific task. Function definition includes function header and function body.

8. What are address operators and indirection operators?
Indirection Operator (*): It accesses a value indirectly through a pointer. The operand must be a pointer value. The result of the operation is the value addressed by the operand.

Address Operator (&): It gives the address of its operand. The operand of the address can be either a function designator or values that designate the object.

9. Define static storage.
Static storage is the one in which a variable persists until the end of the program. Any variable can be declared static using the keyword static. A static variable is initialized only once when a program is compiled. It is never initialized again.

10. What is the use of #define preprocessor?
    #define is a macro which defines a constant value and can be any of the basic data types.

PART–B (5 × 16 = 80 marks)

11. (a) (i) Explain in detail with neat diagram about the digital computer organization and each of its units. (10)
(Refer 2.1 & 2.2)

(ii) What is pseudo code? Write a pseudo code for swapping two numbers without using temporary storage. (6)
(Refer 5.15)

Or

(b) (i) Perform the following: (12)

(1) $(100101)_2$ - $(11111001)_2$

SOLUTION:

```
0 0 1 0 0 1 0 1
            (–)
1 1 1 1 1 0 0 1
1 1 0 1 0 1 0 0
```

Ans.: $(11010100)_2$

(2) $(1011101)_2 \times (1011)_2$

```
      1 0 1 1 1 0 1
                  ×
            1 0 1 1
1 1 1 1 1 1 1 1 1 1
```

Ans.: $(1111111111)_2$

(3) $(2A947)_H = (\ ?\ )_2$

Use 8 4 2 1 method for each split up

$(0\,0\,1\,0 \quad 1\,0\,1\,0 \quad 1\,0\,0\,1 \quad 0\,1\,0\,0 \quad 0\,1\,1\,1)_2$

(4) $(4872)_{10} = (\ ?\ )_8$

```
8    4872
8     609    0
8      76    1
8       9    4
        1      1
```

Ans.: $(11410)_8$

(ii) Discuss the need for the logical analysis with an example in brief? (4)
(Refer 4.1)

12. (a) What are the various operators available in C? Discuss each one of them with suitable illustrations. (16)

(Refer chapter 8)

Or

(b) Explain in detail about the various looping structures available in C with illustrative programs.

(Refer chapter 11)

13. (a) (i) Write a C program for sorting array of numbers. (8)

(Refer 12.7)

(ii) Explain the various string operations. Write a C program to find out the length of the string without using built-in function. (8)

(Refer 13.6) and (Refer 13.7)

Or

(b) (i) Write a C program to multiply two matrices. (8)

(Refer 12.16)

(ii) Write a C program to search an element in the given array. (8)

(Refer 12.8)

14. (a) What is a function in C? Discuss about the call by value and call by reference with illustrations. (16)

(Refer 14.10)

Or

(b) What is recursion? Explain a recursive function with suitable example. Write an iterative and recursive function to find the power of a number. (16)

(Refer 14.13)

```
Program
#include<stdio.h>
int power(int n1,int n2);    //global function declaration
int main()
{
    int base, exp;
    printf(“Enter base value: “);
    scanf(“%d”,&base);
    printf(“Enter the  number: “);
    scanf(“%d”,&exp);
    printf(“%d ^ %d = %d”, base, exp, power(base, exp));
    return 0;
}
int power(int base,int exp)
{
    if ( exp!=1 )
        return (base*power(base,exp-1));
}
```

```
Output
Enter base number: 3
Enter the number 3
3^3 = 27
```

15. (a) (i) What is a structure? Create a structure with data members of various types and declare two structure variables. Write a program to read data into these and print the same. (10)

(Refer chapter 16) & (Refer Appendix-28)

(ii) Justify the need for structured data type. (6)

- Structure can support complex data type to store the different types of variables.
- It is possible to copy the contents of all the structure elements to another structure variable of its type using assignment operator.
- Nesting of structure is possible.
- It is possible to pass complex data type (structure elements) to a function.
- It is possible to create structure pointers.

Or

(b) Write short notes on: (4 × 4 = 16)

(i) Unions **(Refer 16.15)**

(ii) Register storage class **(Refer 17.5)**

(iii) #include statement **(Refer 17.7)**

(iv) #ifndef…#endif **(Refer 17.8)**

B.E/B. Tech. Degree Examination, April/May 2015
GE6151 – Computer Programming
(Common to all branches)
(Regulation 2013)

Time : Three hours **Maximum : 100 Marks**

PART– A (10 × 2 = 20 marks)

1. What are Super Computers? Give an example.
Super Computers are the special purpose machines, which are specially designed to maximize the number of floating point operations per second (FLOPS). A super computer has the highest processing speed at a given time for solving scientific and engineering problems.

2. Define pseudo code.
Pseudo code is made up of two words: pseudo and code. It means imitation and the code refers to instructions written in a programming language. Pseudo code is an outline of a program. It uses plain English statements rather than symbols. It is also known as PDL (Program Design Language).

3. What is the importance of keywords in C?

Keywords are reserved words that have standard and predefined meaning in C language. They are the basic building blocks for program statements. Some of the keywords are:

int, break, continue, for, float, char etc.

4. List the various input and output statements in C.
Input statement: scanf()
Output statement: printf()

5. What is an array? Give an example.
An array is a derived data type. It is a collection of data elements of similar data type under a common name. Each element in the collection is identified by an index.
Example: int a [ ] = {1,2,3,4};

6. How is a character string declared?
char name [ ] ={” st.”, “josephs”,”institute”};

7. Compare actual parameters and formal parameters.

Actual arguments The arguments that are passed in a function call are called actual arguments. These arguments are defined in the calling function.

Formal arguments The formal arguments are the parameters/arguments in a function declaration. The scope of formal arguments is local to the function definition in which they are used.

A change in formal arguments would not be reflected in the actual arguments.

8. What is the output of the following program?

```
main( )
{
int a=8, b=4, c, *p1=&a, *p2=&b;
C=*p1**p2-*p1/*p2+9;
Printf("%d",c);
}
```

Output: compilation error

9. What do you mean by structures?

A structure is a collection of one or more variables of different data types, grouped together under a single name. By using structures, we can make a group of variables, arrays, pointers etc.

10. Give the uses of preprocessor.

Before a C program is compiled in a compiler, source code is processed by a program called preprocessor. This process is called preprocessing.

Commands used in preprocessor are called preprocessor directives and they begin with "#" symbol.

PART– B (5 × 16 = 80 marks)

11. (a) (i) Describe various generations of computers. (10)

Refer 1.6

(ii) Convert the decimal number 681.75 into binary,octal and hexadecimal equivalent. (6)

Binary

2	**681 Decimals**	
2	340	1 ↑
2	170	0
2	85	0
2	42	1
2	21	0
2	10	1
2	5	0
2	2	1
1	0	

0.75 × 2 = 1.50

0.50 × 2 = 1.00

$(1010101001.11)_2$

Octal

8	681	
8	85	1
8	10	5
	10	2

Decimal 0.75 × 8 = 6.00

$(1251.6)_8$

Hexadecimal

16	681	
16	42	9
16	2	10 (A)

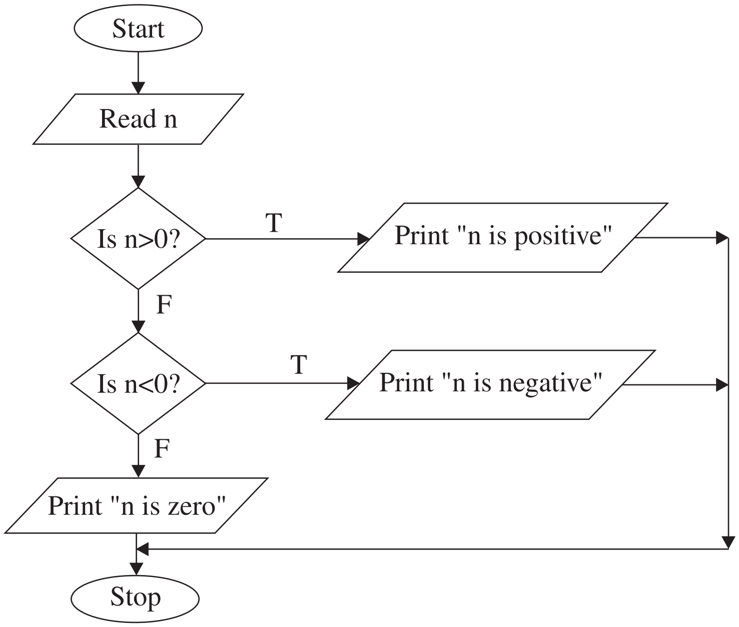
Decimal 0.75 × 16 = 12.00 (12=C)

$(2\,A\,9.C)_{16}$

Or

(b) (i) Explain the basic organization of a computer with neat diagram. (10)
Refer 2.1

(ii) Draw a flowchart to check whether the given number is zero, positive or negative. (6)



12. (a) (i) Explain the different types of operators available in C. (8)

Refer Chapter 8

(ii) What are constants? Explain the various types of constants in C. (8)

Refer 7.4

Or

(b) (i) Describe the various looping statements used in C with suitable examples. (8)

Refer Chapter 11

(ii) Write a C program to solve a quadratic equation. (8)

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
float  a, b, c, d, root1, root2;
clrscr();
printf("Enter the values of a, b, c\n");
scanf("%f%f%f", &a, &b, &c);
if(a == 0 || b == 0 || c == 0)
  {
printf("Roots can't be determined");
  }
else
  {
   d = (b * b) - (4.0 * a * c);
if(d > 0.00)
   {
printf("Roots are real and distinct \n");
       root1 = -b + sqrt(d) / (2.0 * a);
       root2 = -b - sqrt(d) / (2.0 * a);
printf("Root1 = %f \nRoot2 = %f", root1, root2);
   }
else if (d < 0.00)
   {
printf("Roots are imaginary");
       root1 = -b / (2.0 * a) ;
       root2 = sqrt(abs(d)) / (2.0 * a);
printf("Root1 = %f  +i %f\n", root1, root2);
printf("Root2 = %f  -i %f\n", root1, root2);
   }
else if (d == 0.00)
   {
```

```
printf("Roots are real and equal\n");
       root1 = -b / (2.0 * a);
       root2 = root1;
printf("Root1 = %f\n", root1);
printf("Root2 = %f\n", root2);
    }
   }
getch();
}
```

13. (a) (i) Write a C program to add two matrices. (10)

Refer 12.14

(ii) Write a C program to search a given number in an array of elements. (6)

Refer 12.8

Or

(b) (i) Write a C program to arrange the given 10 numbers in ascending order. (10)

Refer 12.7

(ii) Explain the various strings handling functions. (6)

Refer 13.6

14. (a) (i) Write a C program to find the factorial of a given number using function. (8)

Appendix - Ex. 7

(ii) Write a C program to exchange the values of two variables using call by reference. (8)

Refer 14.11

Or

(b) (i) Write a C program to find the sum of the digits using recursive function. (8)

```
#include<stdio.h>
void main(){
intnum,x;
clrscr();
printf("\nEnter a number: ");
scanf("%d",&num);
  x=sum(num);
printf("Sum of the digits of %d is: %d",num,x);
  }

intr,s;
int sum(int n)
{
if(n){
        r=n%10;
        s=s+r;
 sum(n/10);
      }
 else
 return s;
 }
```

(ii) Write a C program using pointers to read in an array of integers and print its element in reverse order. (8)

```
#include<stdio.h>
#include<conio.h>
#define MAX 30

void main() {
  int size, i, arr[MAX];
  int *ptr;
  clrscr();

  ptr = &arr[0];

  printf("\nEnter the size of array : ");
  scanf("%d", &size);

  printf("\nEnter %d integers into array: ", size);
  for (i = 0; i< size; i++) {
    scanf("%d", ptr);
    ptr++;
  }

  ptr = &arr[size - 1];

  printf("\nElements of array in reverse order are :");

  for (i = size - 1; i>= 0; i--) {
    printf("\nElement%d is %d : ", i, *ptr);
    ptr--;
  }

  getch();
}
```

15. (a) Define a structure called book with book name,author name and price . Write a C program to read the details of the book name,author name and price of 200 books in a library and display the total cost of the books and the book details whose price is above Rs. 500. (16)

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
void book_auth();
void book_pub();
void book_price();
void disp_book();
void add_book();

struct book
{
char title[20];
int  acc_no;
char author[20];
char pub[20];
intprice;
};
int count;
struct book b[100];

void main()
{
intch;
while(1)
{
clrscr();
printf(“\n 1:add book \n”);
printf(“\n 2:specific author \n”);
printf(“\n 3:specific publisher \n”);
printf(“\n 4:price above 500 \n”);
printf(“\n 5:all book \n”);
printf(“\n 6:exit \n”);
printf(“\n\n enter the choice \n”);
scanf(“%d”,&ch);
switch(ch)
{
case 1: add_book(); getch();
break;
case 2: book_auth(); getch();
break;
case 3: book_pub(); getch();
break;
case 4: book_price(); getch();
break;
case 5: disp_book(); getch();
break;
case 6:exit(0);
}
}
}
void add_book()
{
if(count==9)
{
printf("\n no more space");
return;
}
printf("\n enter book detail ");
printf("\n enter accession number of book \n");
scanf("%d",&b[count].acc_no);
printf("\n enter name of book\n");
scanf("%s",b[count].title);
printf("\n enter name of author\n");
scanf("%s",b[count].author);
printf("\n enter the publisher\n");
scanf("%s",b[count].pub);
printf("\n enter book price \n");
scanf("%d",&b[count].price);
count++;
}
void disp_book()
{
inti;
printf("\n detail of %d book ",count);
for(i=0;i<count;i++)
{
printf("\n %d\n%s\n%s\n%s\n%d",b[i].acc_no,b[i].title,b[i].author,b[i].pub,b[i].price);
}
}

void book_auth()
```

```
{
inti,cnt;
char name[20];
printf("\n enter the name of author");
scanf("%s",name);
for(i=0;i<count;i++)
{
if(strcmp(name,b[i].author)==0)
{
cnt++;
printf("\n%d\n%s\n%s\n%s\n%d",b[i].acc_no,b[i].title,b[i].author,b[i].pub,b[i].price);
}
if(cnt==0)
printf("\n no such book");
}
}
void book_pub()
{
inti,cnt;
char name[20];
printf("\n enter the name of publisher=");
scanf("%s",name);
for(i=0;i<count;i++)
{
if(strcmp(name,b[i].pub)==0)
{
cnt++;
printf("\n%d\n%s\n%s\n%s\n%d",b[i].acc_no,b[i].title,b[i].author,b[i].pub,b[i].price);
}
}
if(cnt==0)
printf("\n no such book");
}
void book_price()
{
inti;
for(i=0;i<count;i++)
{
if(b[i].price>500)
{
printf("\n%s",b[i].title);
```

```
        }
    }
}
```

Or

(b) (i) Explain the various storage classes in C. (10)

Refer 17.1

(ii) What is union? Discuss with an example. (6)

Refer 16.15

B.E/B.Tech Degree Examination, Nov/Dec 2015
FIRST SEMESTER
GE6151 – Computer Programming
(Common to all branches)
(Regulation 2013)

Time : Three hours **Maximum : 100 Marks**

Answer ALL questions

PART–A (10 × 2 = 20 marks)

1. What is pseudo code?
Pseudo code is made up of two words: pseudo and code. It means imitation and the code refers to instructions written in a programming language. Pseudo code is an outline of a program. It uses plain English statements rather than symbols. It is also known as PDL (Program Design Language).

2. What is an algorithm?
Algorithms are one of the most basic tools that are used to develop the problem solving logic. An algorithm is defined as a finite sequence of explicit instructions that when provided with a set of input values processes an output and then terminates. Algorithms can have steps that repeat or require decisions until the task is completed.

3. What is compilation process?
During the compilation process, the source program instructions are translated into a form that is suitable for execution by the computer. The translation process checks each and every instruction for correctness and if no errors are reported then it generates the object code.

4. Discuss the working of modulo operator.
Modulo operator is used to return the remainder.

5. Declare a character array of size 5 and assign vowels to it.

```
char   vowel[5] = {‘a’, ‘e’, ‘i’, ‘o’, ‘u’, ‘\0’};
```

6. Give some examples of string functions.

Strlen(),strcpy(),strcmp(),strcat(),strchr().

7. What is function definition?
A function definition is an independent program that performs a specific task. Function definition includes function header and function body. Function body contains list of instructions enclosed within a pair of brackets.

The general syntax of function definition:

```
<return-type> fun-name(argument/parameter list)
```

```
{
 // body of the function
}
```

8. What are address operators and indirection operators?

Indirection operator: The indirection operator (*****) accesses a value indirectly through a pointer. The operand must be a pointer value. The result of the operation is the value addressed by the operand, that is, the value at the address to which its operand points. The type of the result is the type that the operand addresses

Address operator: The address operator (**&**) gives the address of its operand. The operand of the address operator can be either a function designator or l-value that designates an object that is not a bit field and is not declared with the **register** storage-classspecifier.

9. Write a note on register storage class.
Registers are able to store only restricted size value, so most of the compilers will allow only int and char variables to be placed in the register. Registers are limited in count, therefore it is important to select required variables for this purpose. The C compiler automatically converts register variables into non register variables once the limit is reached,

Syntax:

```
register  int count;
```

10. What is the use of #define preprocessor?
#define macro defines constant value and can be any of the basic data types.

PART– B (5 × 16 = 80)

11. (a) Explain in detail with block diagram about the digital computer organization and discuss the function of each block. (16)
Refer 2.1

Or

(b) Perform the following: (4 × 4 =16)

(i) $\mathbf{(1011.11011)_2 = (\ \ )_{10}}$

$2^3\,2^2\,2^1\,2^0.\,2^{-1}\,2^{-2}\,2^{-3}\,2^{-4}\,2^{-5}$

1 0 1 1 . 1 1 0 1 1

8+0+2 +1 . 0.5+0.25+0+0.0625+0.03125

$\mathbf{(11.84375)_{10}}$

(ii) $\mathbf{(10111)_2 \times (1011)_2 = ?}$

$$\begin{array}{r} 1\,0\,1\,1\,1 \\ \times \\ 1\,0\,1\,1 \\ \hline \mathbf{1\,1\,1\,1\,1\,1\,0\,1} \end{array}$$

(iii) $(\mathrm{D8BC})_H = (\ ?\ )_2$

$(1101\ \ 1000\ 1011\ \ 1100)_2$

(iv) $(4871)_{10} = (\ ?\ )_8$

$$\begin{array}{r|ll}
8 & 4871 & \\ \cline{2-2}
8 & 608 & 7 \\ \cline{2-2}
8 & 76 & 0 \\ \cline{2-2}
8 & 9 & 4 \\ \cline{2-2}
 & 1 & 1
\end{array}$$

12. (a) What are the various operators available in C? Discuss each one of them with suitable examples? (16)

Refer Chapter 8

Or

(b) Explain in detail about various decision making structures available in C with illustrative examples. (16)

Refer Chapter 10

13. (a) Write a C program for finding the largest element and smallest element in a matrix. (16)

```
#include <stdio.h>
int main()
{
   int arr[100];
   int i, max, min, size;
   printf(“Enter size of the array: “);
   scanf(“%d”, &size);
   printf(“Enter elements in the array: “);
   for(i=0; i<size; i++)
   {
      scanf(“%d”, &arr[i]);
   }
   max = arr[0];
   min = arr[0];
   for(i=1; i<size; i++)
   {
         if(arr[i]>max)
      {
            max = arr[i];
        }
         if(arr[i]<min)
        {
            min = arr[i];
        }
    }
    printf("Maximum element = %d\n", max);
    printf("Minimum element = %d", min);

    return 0;
}
```

Or

(b) Write a C program to multiply two matrices. (16)

Refer 12.16

14. (a) Discuss about call by value and call by reference with illustrations. (16)

Refer 14.10

Or

(b) What is recursion? Explain a recursive function with suitable example. Write a recursive function to find the factorial of a number. (16)

Refer 14.13

```
#include<stdio.h>
int fact(int);
void main(){
 int num,f;
 printf("\nEnter a number: ");
 scanf("%d",&num);
 f=fact(num);
 printf("\nFactorial of %d is: %d",num,f);
  }

int fact(int n){
  if(n==1)
     return 1;
  else
     return(n*fact(n-1));
 }
```

15. (a) What is a structure? Create a structure with data members of various types and declare two structure variables. Write a program to read data in to these and print the same. (16)
Refer 16.1

Or

(b) Write short notes on: (4 × 4 =16)

(i)	Union	**Refer 16.15**
(ii)	Static storage class	**Refer 17.4**
(iii)	#include statement	**Refer 17.7**
(iv)	#ifndef … #endif	**Refer 17.8**