4044(15) (1)

| Qst No | Rev(15) 4044 PROGRAMMING IN C | | |
|---|---|---|---|
| I | **PART A** | | |
| 1 | • & (bitwise AND)<br>• \| (bitwise OR)<br>• ^ (bitwise XOR)<br>• << (left shift) | 2X1 | 2 |
| 2 | do {<br>statement(s);<br>} while( condition ); | | 2 |
| 3 | A pointer is a variable whose value is the address of another variable, i.e., direct address of the memory location. Like any variable or constant, you must declare a pointer before using it to store any variable address. | | 2 |
| 4 | strcmp (s1, s2); Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2. | 1+1 | 2 |
| 5 | Global variables are defined outside a function. Global variables hold their values throughout the lifetime of the program.A global variable can be accessed by any function. | | 2 |
| II | **PART B** | | |
| 1 | • All variable names must begin with a letter of the alphabet or an. underscore( _ )<br>• No spaces are allowed in variable declaration.<br>• After the first initial letter, variable names can also contain letters and numbers<br>• Uppercase characters are distinct from lowercase characters<br>• You cannot use a keyword (reserved word) as a variable name.<br>• Except underscore (_) no other special symbol are allowed. | 4X1.5 | 6 |
| 2 | SAMPLE PROGRAM<br>#include <stdio.h><br>int main() {<br>int a, b, largest;<br>printf("Please Enter Two Different Values\n");<br>scanf("%d %d", &a, &b);<br>largest = (a > b) ? a : b;<br>printf("%d is Largest\n", largest);<br>return 0;<br>} | | 6 |
| 3 | #include <stdio.h><br>int main()<br>{<br>int array[100], position, c, n;<br>printf("Enter number of elements in array\n"); | | 6 |

| | | | |
|---|---|---|---|
| | scanf("%d", &n);<br>printf("Enter %d elements\n", n);<br>for (c = 0; c < n; c++)<br>scanf("%d", &array[c]);<br>printf("Enter the location where you wish to delete element\n");<br>scanf("%d", &position);<br>for (c = position - 1; c < n - 1; c++)<br>array[c] = array[c+1];<br>printf("Resultant array:\n");<br>for (c = 0; c < n - 1; c++)<br>printf("%d\n", array[c]);<br>}<br>return 0;<br>} | | |
| 4. | Syntax of a For Loop: for(initialisation;condition;update) eg:-for(i=0;i<10;i++)<br><br>Any simple example of a For Loop with explanation | 2<br><br>4 | 6 |
| 5. | There are four arithmetic operators that can be used on pointers: ++, --, +, and –<br><br>Explain with Simple examples of any three. | 2X3 | 6 |
| 6. | The subprogram are easier to write, understand and debug.function can call itself again. It is called a recursive function.Reduction in size of a program.<br>The complexity of the entire program can be divided into simple subtask. | 4X1.5 | 6 |
| 7. | Recursion, i.e., a function to call itself.The following sample program calculates the factorial of a given number using a recursive function<br>#include <stdio.h><br>unsigned long long int factorial(unsigned int i) {<br>if(i <= 1) {<br>return 1;<br>}<br>return i * factorial(i - 1);<br>}<br>int main() {<br>int i = 12;<br>printf("Factorial of %d is %d\n", i, factorial(i));<br>return 0;} | 2<br><br>4 | 6 |
| III | **PART C** | | |
| (a) | C has the following basic built-in datatypes:-Int,float,double and char<br><br>int is used to define integer numbers eg:- int x,y;x=5;y=10;<br><br>float is used to define real numbers. Eg:-float sum=9.7; | 2X4 | 8 |

double is used to define big real numbers. twice the storage. Eg;-double sum;

char defines characters. Eg:-char letter ='x';

The data types explained above have the following modifiers;-short,long,signed and unsigned

The modifiers define the amount of storage allocated to the variable. ANSI has the following rules:

short int <= int <= long int
float <= double <= long double

b)

```
// C Program to Check Vowel or Consonant
#include <stdio.h>
int main()
{
   char ch;
   printf("Please Enter an alphabet: \n");
   scanf(" %c", &ch);
   if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
                ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')  {
                printf("\n %c is a VOWEL.", ch);
        }
   else {
        printf("\n %c is a CONSONANT.", ch);
        }
   return 0;
```

7

IV a)

**BASIC STRUCTURE OF A 'C' PROGRAM:**

| | Example: |
|---|---|
| **Documentation section** [Used for Comments] | //Sample Prog Created by:Bsource |
| **Link section** | #include<stdio.h><br>#include<conio.h> |
| **Definition section** | void fun(); |
| **Global declaration section** [Variable used in more than one function] | int a=10; |
| **main()**<br>{<br>**Declaration part**<br>**Executable part**<br>} | void main()<br>{<br>clrscr();<br>printf("a value inside main(): %d",a);<br>fun();<br>} |
| **Subprogram section** [User-defined Function]<br>**Function1**<br>**Function 2**<br>:<br>:<br>**Function n** | void fun()<br>{<br>printf("na value inside fun(): %d",a);<br>} |

4+4

8

b)

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int a,b;
   int op;
   printf("Enter the values of a & b: ");
   scanf("%d %d",&a,&b);
   printf("Enter your Choice : ");
   scanf("%d",&op);
   switch(op)
   {
   case 1  :
      printf("Sum of %d and %d is : %d",a,b,a+b);
      break;
   case 2  :
      printf("Difference of %d and %d is : %d",a,b,a-b);
      break;
   case 3  :
      printf("Multiplication of %d and %d is : %d",a,b,a*b);
      break;
   case 4  :
      printf("Division of Two Numbers is %d : ",a/b);
      break;
   default :
      printf(" Enter Your Correct Choice.");
      break;
   }
   getch();
}.
```

7

V

a)

| Entry Controlled Loop | Exit Controlled Loop |
|---|---|
| Test condition is checked first, and then loop body will be executed | Loop body will be executed first, and then condit is checked. |
| If Test condition is false, loop body will not be executed. | If Test condition is false, loop body will be execu once. |
| for loop and while loop are the examples of Entry Controlled Loop. | do while loop is the example of Exit controlled lo |
| Entry Controlled Loops are used when checking of test condition is mandatory before executing loop body. | Exit Controlled Loop is used when checking of te condition is mandatory after executing the loop b |

4x2

8

b) Sample program

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

```
void main()
{
 int a[10][10],b[10][10],i,j;
 printf("\nEnter the elements of matrix:");
 for(i=0;i<3;i++)
{
 for(j=0;j<3;j++)
{
 scanf("%d",&a[i][j]);
}
}
 printf("\nThe Transpose of matrix is:\n");
 for(i=0;i<3;i++)
{
 for(j=0;j<3;j++)
{
 b[i][j]=a[j][i];
 printf("%d",b[i][j]);
}
}
 getch();
}
```

7

VI

a) C Array is a collection of variables belongings to the same data type. You can store group of data of same data type in an array.Below shows a sample program.

```
#include<stdio.h>
int main()
{
    int i;
 int arr[5] = {10,20,30,40,50};// declaring and Initializing array in C
      /* Above array can be initialized as below also
      arr[0] = 10; arr[1] = 20; arr[2] = 30; arr[3] = 40; arr[4] = 50; */
    for (i=0;i<5;i++)  {
       // Accessing each variable
       printf("value of arr[%d] is %d \n", i, arr[i]);
     }
}
```

8

b)

```
/*Sample program factorial of a number*/

#include<stdio.h>
#include<conio.h>
void main()
{
   int n,i,f;
   f=i=1;
   clrscr();
   printf("Enter a Number to Find Factorial: ");
   scanf("%d",&n);
```

7

```
    while(i<=n)
    {
       f*=i;
       i++;
    }
    printf("The Factorial of %d is : %d",n,f);
    getch();
}
```

VII a) Explain any four with simple examles: 4x2 — 8

| | |
|---|---|
| strlen(str1) | It returns the length of string str1 |
| strlwr(str1) | It returns all the uppercase letters to lowercase letters. |
| strupr(str1) | It returns all the lowercase letters to uppercase letters. |
| strcat(str1,str2) | It is used to concatenate two strings str1 and str2 |
| strcpy(str1,str2) | It copies str2 to str1 |
| strcmp(str1,str2) | Returns 0 if str1 and str2 are the same; less than 0 if str1<str2; greater than 0 if str1>str2 |

b) /*reverse the given string using pointers*/ — 7

```
#include <stdio.h>
#include <conio.h>
void main()
{
        char *s;
        int len,i;
        clrscr();
        printf("\nENTER A STRING: ");
        gets(s);
        len=strlen(s);
        printf("\nTHE REVERSE OF THE STRING IS:");
        for(i=len;i>=0;i--)
                printf("%c",*(s+i));
        getch();
}
```

VIII a) A pointer is a variable whose value is the address of another variable, i.e., direct address of the memory location. Like any variable or constant, you must declare a pointer before using it to store any variable address. The general form of a pointer variable declaration is – type *varname; — 8

```
#include <stdio.h>

int main () {

  int  var = 20;   /* actual variable declaration */

  int  *ip;        /* pointer variable declaration */
```

```
   ip = &var;  /* store address of var in pointer variable*/
   printf("Address of var variable: %x\n", &var  );
   /* address stored in pointer variable */
   printf("Address stored in ip variable: %x\n", ip );
   /* access the value using the pointer */
   printf("Value of *ip variable: %d\n", *ip );
   return 0;}
```

VIII b)

```
/*Palindrome program in C language*/

#include <stdio.h>
#include <string.h>
 int main(){
  char a[100], b[100];
  printf("Enter a string to check if it is a palindrome\n");
  gets(a);
  strcpy(b, a);  // Copying input string
  strrev(b);  // Reversing the string
  if (strcmp(a, b) == 0)  // Comparing input string with the reverse string
    printf("The string is a palindrome.\n");
  else
    printf("The string isn't a palindrome.\n");
  return 0;}
```

7

IX a)

1.Function without arguments and no return values
2. Function without arguments and return values
3. Function with arguments and no return values
4. Function with arguments and return values
Explain each with simple programs.

4x2

8

b)

```
/* Sample C Program to Find Sum of Digits of a Number Using Functions */

#include <stdio.h>
int Sum_Of_Digits (int);
int main(){
  int Number, Sum = 0;
  printf("\n Please Enter any number\n");
  scanf("%d", &Number);
  Sum = Sum_Of_Digits (Number);
  printf("\n Sum of the digits of Given Number = %d", Sum);
  return 0;
}
int Sum_Of_Digits (int Number){
  int Reminder, Sum;
```

7

```
  for (Sum=0; Number > 0;Number=Number/10) {
    Reminder = Number % 10;
    Sum=Sum+ Reminder;  }
 return Sum;}
```

X

a)

The call by value method of passing arguments to a function copies the actual value of an argument into the formal parameter of the function. In this case, changes made to the parameter inside the function have no effect on the argument. 4

```
void swap(int x, int y) {
  int temp;
  temp = x; /* save the value of x */
  x = y;   /* put y into x */
  y = temp; /* put x into y */
  return;}
```

The call by reference method of passing arguments to a function copies the reference of an argument into the formal parameter. Inside the function, the reference is used to access the actual argument used in the call. This means that changes made to the parameter affect the passed argument.To pass the value by reference, argument reference is passed to the functions just like any other value. 8

4

```
void swap(int &x, int &y) {
  int temp;
  temp = x; /* save the value at address x */
  x = y;   /* put y into x */
  y = temp; /* put x into y */
  return;}
```

b)

```
/* C Program to Pass Array to Functions  */

#include<stdio.h>
int SumofElements(int a[]){
   int i,sum = 0;
   for(i = 0; i < 10; i++)
      sum = sum + a[i];
   return sum;
}
int main(){
   int i,sum, a[10];
   printf("\nPlease Enter Array Elements\n");
   for(i = 0; i < 10; i++){
      scanf("%d", &a[i]);
   }
   sum = SumofElements(a);
   printf("Sum of All Elements in an Array = %d \n", sum);
   return 0;
}
```

7