

B. Tech.

SECOND SEMESTER EXAMINATION, 2008-09

Computer Concepts & Programming In C

Time : 3 Hours]

[Total Marks : 100

Note : (i) The question paper contains three sections. **Section-A** contains objective type questions. **Section-B** contains short answer type questions and **Section-C** contains descriptive types questions.

(ii) Attempt all questions.

Section-A

Note : This section contains 20 objective type questions. Answer all questions. $20 \times 1 = 20$

Q. 1. Multiple choice

(i) The binary code of $(73)_{10}$ is

- (a) 1010001 (b) 1000100
(c) 1100101 (d) 1001001
(e) 0110110

Ans.

2	73 - 1
2	36 - 0
2	18 - 0
2	9 - 1
2	4 - 0
2	2 - 0
	1

$(1001001)_2$.

(ii) Which of the following is not an input device ?

- (a) Light pen (b) MICR
(c) Microphone (d) Speaker

(e) None of the above.

Ans. (d) Speaker.

(iii) A block is enclosed within a pair of

- (a) { } (b) ()
(c) [] (d) < >
(e) None of the above.

Ans. (a) { }

(iv) The feature which permits a user to execute multiple programs simultaneously ?

- (a) Multi-user
(b) Multi-tasking
(c) Multi-process
(d) Multi-programming
(e) None of the above.

Ans. (d) Multi-programming.

(v) In which of the following loops the number of iterations remain known prior to the execution of the loop ?

- (a) While
(b) Do-while
(c) For
(d) In none of the above.

Ans. (c) For.

(vi) Which operator is used for accessing the members of a structure ?

- (a) dot (.)
(b) arrow (\rightarrow)
(c) Both of the above
(d) None of the above.

Ans. (a) dot (.).

(vii) Which of the following is not a data structure ?

- (a) queue (b) stack
(c) tree (d) variable

Ans. (d) variable.

(viii) Which character terminates strings in C?

- (a) /0 (b) \0
(c) '\0' (d) !0
(e) None of the above.

Ans. (a) /0.

(ix) Which of the following is correct statement for checking a condition in if statement?

- (a) if (a = b) (b) if {a, b}
(c) if (a b) (d) if (a == b)
(e) None of the above.

Ans. (a) if (a == b).

(x) At which stage 'C' preprocessor works?

- (a) Before translation
(b) Before execution
(c) Before linking
(d) Before debugging
(e) None of the above.

Ans. (a) Before translation.

State True/False

(xi) The do while statement first executes the loop body and then evaluate the loop control expression.

Ans. True.

(xii) C functions can return only one value under their function name.

Ans. True.

(xiii) During the evaluation of mixed expression, an implicit cast is generated automatically.

Ans. True.

(xiv) All static variables are automatically initialized to zero.

Ans. True.

(xv) A struct type in C is a built in data types.

Ans. False.

Fill in the blanks :

(xvi) The header file contains mathematical functions.

Ans. math.h

(xvii) is used to determine the order in which different operators in an expression are evaluated.

Ans. Priority

(xviii) is the process for arranging the elements of an array in order.

Ans. Sorting

(xix) A variable can be made constant by declaring it with the qualifier at the time of initialization.

Ans. const

(xx) The parameters used in a function call are called

Ans. arguments.

Section-B

Note : Attempt any three parts from this section. $10 \times 3 = 30$

Q. 2. (a) (i) Area of a triangle is given by the formula

$$A = \sqrt{S(S-a)(S-b)(S-c)}$$

where a , b and c are sides of the triangle and $S = (a + b + c) / 2$. Write a 'C' program to compute the area of the triangle given the values of a , b and c .

Ans. #include <stdio.h>

#include <conio.h>

#include <math.h>

main ()

{

float a, b, c, z;

float area (float a, float b, float c);

clrscr ();

```

printf ("\n Enter three sides of the triangle :");
scanf ("%f%f%f", &a, &b, &c);
z = area (a, b, c);
printf ("\n\n Area of the triangle = %3f", z);
getch ( );
}

float area (float a, float b, float c)
{
float s, m, x;
s = (a + b + c)/2;
m = s* (s - a)* (s - b)* (s - c);
x = sqrt (m);
return (x);
}

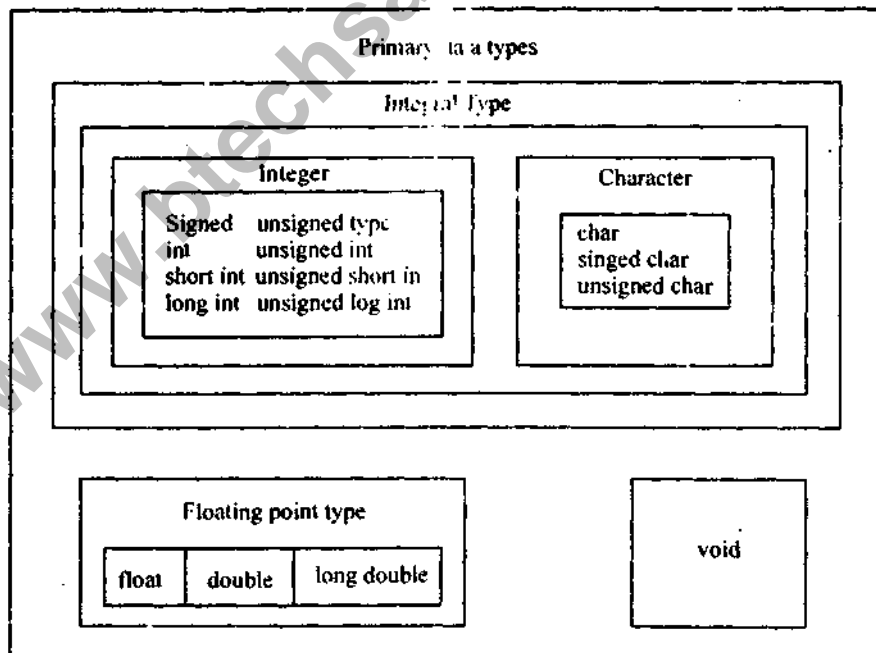
```

Q. 2. (a) (ii) Explain different data types in 'C'.

Ans. C language is rich in its data types. The variety of data types available allow the programmer to select the type appropriate to the needs of the application as well as the machine.

ANSI C supports three classes of data type.

1. Primary data types
2. Derived data types
3. User defined data types.



Q. 2. (b) (i) Explain differences between precedence and associativity of operators with suitable example.

Ans. 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 the associativity property of an operator.

Q. 2. (b) (ii) Write a 'C' program to test whether a given year is leap year or not.

```

Ans. # include <stdio.h>
        # include <conio.h>
        main ( )
        {
            int yr;
            clrscr ( );
            printf ("Enter a year");
            scanf ("%d", &yr);
            if (yr% 100 == 0)
            }
            if (yr% 400 == 0)
            printf ("\n Leap year");
        else
            printf ("\n Not a Leap
year");
    }
    else
    {
        if (yr%4 == 0)
            printf ("In Leap Year");
        else
            printf ("\n Not a Leap
Year");
    }
    printf ("\n\n\n\n\n
press any key to exit");
    getch ( );
}

```

Q. 2. (c) (i) Explain difference between a flow chart and an algorithm with an example.

Ans. An algorithm is just a detailed sequence of simple steps that are needed to solve a problem. The word sequence implies there is an order to the steps.

A flow chart is a graphical representation of an algorithm.

Algorithm may be presented :

1. in words
2. as a flowchart
3. in structured code.

A computer program is an algorithm written for a computer in a special programming language.

A flowchart is a common type of chart, that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting these with arrows. Flowcharts are used in analyzing, designing, documenting and managing a process or program in various fields.

Q. 2. (c) (ii) Write a program in 'c' to accept 10 integer numbers and print them in reverse order.

```

Ans. # include <stdio.h>
        # define SIZE 10
        int main (void)
        {
            int x, numbers [SIZE];
            for (x = 0; x < SIZE; x++)
            }
            printf ("Enter numbers
:");
            scanf ("%d", & numbers [x]);
        }
        for (x = SIZE -1; x >= 0; x --)
            printf ("%d\n", numbers
[x]);

```

Q. 2. (d) (i) Write a program in 'C' to receive an integer and find its octal equivalent.

```

Ans. # include <stdio.h>
# include <conio.h>
# include <math.h>
main ( )
{
int n1, dg, oct, p;
printf ("\n Enter any number :");
scanf ("%d", &n1);
p = oct = 0;
while (n1 > 0)
{
dg = n1 % 8;
n1 = n1/8;
oct = oct + dg * pow (10, p);
p++;
}
printf ("The octal equivalent of %d is
%d", n1, oct);
getch ( );
}

```

Q. 2. (d) (ii) Write a C program to generate the following pattern

```

*
* *
* * *
* * * *

```

```

Ans. main ( )
{
int i, j;
for (j = 0; j <= 4; j++)
{
for (i = 0; i <= j; i++)
printf (" * ");

printf ("\n");
}
}

```

Output :

```

*
* *
* * *
* * * *

```

Q. 2. (e) (i) What is a pointer ? Write a 'C' program to swap the values of two variables making use of pointers.

Ans. A pointer is a derived data type in C. It is built from one of the fundamental data types available in C. Pointers contain memory addresses as their values. Since these memory addresses are the locations in the computer memory where program instructions and data are stored, pointers can be used to access and manipulate data stored in the memory.

Pointers are used frequently in C, as they offer a number of benefits to the programmers. They include :

1. Pointers are more efficient in handling array and data tables.
2. Pointers can be used to return multiple values from a function via function arguments.
3. Pointers permit references to functions and thereby facilitating passing of functions as arguments to other functions
4. Pointers reduce length and complexity of programs

```

main ( )
{
int a = 10, b = 20;
swapr (&a, &b);
printf ("\n a = %d, b = %d", a, b);
getch ( );
}

swapr (int *x, int *y)
{
int t;
t = *x;
*x = *y;
*y = t;
}

```

Q. 2. (e) (ii) Write a 'c' program to copy the contents of one file to another.

```

Ans. # include <stdio.h>
# include <conio.h>
main ( )
{
File *fs, *ft;
char ch;
char source [67], target [67];
clrscr ( );
printf ("Enter source file name :");
gets (source);
printf ("Enter target file name :");
gets (target);
fs = fopen (source , "r");
if (fs == NULL)
{
printf ("Unable to open source file.");
getch ( );
exit ( );
}
ft = fopen (target, "wt");
if (ft == NULL)
{
fclose (fs);
puts ("Unable to open target file.");
getch ( );
exit ( );
}
printf ("\n file copied!!");
fclose (fs);
fclose (ft);
getch ( );
}

```

Section-C

Note : There are five questions in the section. Attempt all questions. $10 \times 5 = 50$

Q. 3. Attempt any one part :

Q. 3. (a) Convert the following :

(i) $(478A)_{16} = (X)_{10}$

(ii) $(975 \cdot 55)_{10} = (Y)_2$

(iii) $(11001100 \cdot 10)_2 = (Z)_{16}$

(iv) $(678)_{10} = (W)_3$

(v) $(752)_8 = (V)_{16}$

Ans. (i) $(478A)_{16} = (X)_{10}$

$$3 \times (16)^3 + 7 \times (16)^2 + 8 \times (16)^1 + 10 \times (16)^0$$

$$3 \times 4096 + 1792 + 128 + 10$$

$$12288 + 1792 + 128 + 10$$

$$Y_2 = (14218)_2$$

(ii) $(975 \cdot 55)_{10} = (Y)_2$

2	975 - 1
2	487 - 1
2	243 - 1
2	121 - 1
2	60 - 0
2	30 - 0
2	15 - 1
2	7 - 1
2	3 - 1
	1

1111001111.100011

·55

× 2

1·10 →

·10

× 2

·20 →

× 2

·40 →

× 2

·80 →

× 2

$$1:60 \rightarrow \boxed{1}$$

$$\times 2$$

$$1:2 \rightarrow \boxed{1}$$

$$(iii) (11001100 \cdot 10)_2 = (Z)_{16}$$

$\underbrace{\quad\quad}_c \quad \underbrace{\quad\quad}_c$

$$(Z)_{16} \Rightarrow (CC \cdot 8)_{16}$$

$$(iv) (678)_{10} = W_3$$

3	678 - 0
3	226 - 1
3	75 - 0
3	25 - 1
3	8 - 2
2	2
	1

$$(221010)_3$$

$$(v) (752)_8 = (V)_{16}$$

$$= 1EA$$

$$7 \times 8^2 + 5 \times 8^1 + 2 = (490)_2$$

16	490 10 A
16	30 - 14 → E
	1

$$(1EA)_{16}$$

Q. 3. (b) Write your own 'C' functions to implement the following standard library string functions :

(i) strlen ()

Ans. strlen ()

main ()

{

char arr [] = "Bamboozled";

int len1, len2;

len1 = strlen (arr);

len2 = strlen ("Humpty Dumpty");

printf ("\n string = % s length = % d", arr, len1);

printf ("\n string = % s length = % d", "Humpty Dumpty", len2);

strlen (char *s)

{
int length = 0;
while (%s! = '\0')

{
length ++;
s++;

return (length);

}

The output :

String = Bamboozled length = 10

String = Humpty Dumpty length = 13

(ii) strcpy ()

Ans. (ii) strcpy ()

main ()

{

char source [] = "sayonara";

char target [20];

x strcpy (target, source);

printf ("In source string = % s", source);

print ("In target string = % s", target);

}

strcpy (char *t, char *s)

{

while (*s! = '\0')

{

*t = *s;

s++;

t++;

}

*t = '\0';

}

The output of the program :

Source string = Sayonara

target string = Sayonara

Q. 4. Attempt any one part :

Q. 4. (a) A company insures its drivers in the following cases :

– If the driver is married.

– If the driver is unmarried, male and above 30 years of age.

– If the driver is unmarried and above 25 years of age.

In all others cases, the driver is not insured. Write a 'C' program without using logical operators to determine whether the driver is insured or not.

Ans. main ()

{

char sex, ms;

int age;

printf ("Enter age, sex, marital status");

scanf ("%d %c %c", &age, &sex, &ms);

If ((ms == 'M') || (ms == 'U' && sex == 'M' && age > 30) ||

(ms == 'U' && sex == 'F' && age > 25))

printf ("Driver is insured");

else

printf ("Driver is not insured");

}

Q. 4. (b) Write a 'C' program to add first seven terms of the following series using a for loop :

$$\frac{1}{1} + \frac{2}{2!} + \frac{3}{3!} + \dots$$

Ans. # include <stdio.h>

include <conio.h>

main ()

{

int i = 1, j;

float fact, sum = 0.0;

dscr ();

while (i <= 7)

{

fact = 1.0;

for (j = 1; j <= i; j++)

fact = fact *j;

sum = sum +i/fact;

i++;

}

printf ("\n sum of series =%f", sum);

printf ("\n \n press any key to exit ... ");

getch ();

}

Q. 5. Attempt any one part :

Q. 5. (a) A 5-digit positive interger is entered through the keyboard write a 'C' function to calculate sum of digits of the 5-digit number :

(i) Without using recursion

(ii) Using recursion...

Ans. # include <stdio.h>

include <conio.h>

main ()

{

int s, rs;

int n;

dscr ();

printf ("Enter number");

scanf ("%d", &n);

s = sum (n);

printf ("\n sum digits without using recursion is %d", s);

rs = r sum (n);

printf ("\n \n Sum of digitis using recursion is %d", rs);

getch ();

}

sum (int num) /* Function without recursion */

{

int remainder, sum = 0;

while (num > 0)

{

remainder = num% 10;

sum = sum + remainder;

num = num/10;


```

    }
    return (sum);
}
rsum (int num)/* Function with
recursion*/
{
    int sum = 0;
    int remainder;
    if (num != 0)
    {
        remainder = num %10;
        sum = remainder + rsum (num /10);
    }
    return sum;
}

```

Q. 5. (b) Write macro definition with arguments for calculation of simple interest and amount. Store these macro definitions in a file called 'Interst.h'. Include this file in your program, and use the macro definitions for calculating simple interest and amount.

```

Ans. # include <stdio.h>
# include <conio.h>
# include "Interst.h"
main ()
{
    int p, n;
    float si, amt, r;
    clrscr ();
    printf ("\n Enter principal, no. of years
and rate of interest :");
    scanf ("%d %d %f", &p, &n, &r);
    si = S1 (p, n, r);
    amt = AMT (si, p);
    printf ("Simple interest is %f \n amount is
: %f", si, amt);
    getch ();
}

```

Q. 6. Attempt any one part :

Q. 6. (a) Write a program in 'C' that compares two given dates. To store a date, use

a structure that contains three members namely day, month and year. If the dates are equal then display message as "equal" otherwise "Unequal".

```

Ans. # include <stdio.h>
# include <conio.h>
struct date
{
    int day, month, year;
};
int check_date (struct date *dt);
void main ()
{
    int chkdt;
    struct date d1, d2;
    clrscr ();
    printf ("\n enter the dates to be
compared:");
    chkdt = check_date (&d1);
    if (chkdt == 0)
        exit ();
    fflush (stdin);
    chkdt = check_date (&d2);
    if (chkdt == 0)
        exit ();
    if ((d1.day == d2.day) && (d1.month ==
d2.month) && (d1.year == d2.year))
        printf ("\n dates are equal");
    else
        printf ("\n dates are unequal");
    getch ();
}
int check_date (struct date *)
{
    printf ("\n Enter date (dd) :");
    scanf ("%d", &dt -> day);
    printf ("Enter month (mm) : ");
    scanf ("%d", &dt -> month);
    printf ("\n Enter year (yyyy) :");
    scanf ("%d", &dt -> year);
}

```

```

if ((dt -> day > 31 || dt -> day < 0) ||
(dt -> month > 12 || dt -> month < 0) ||
(dt -> year > 9999 || dt -> year < 1000))
{
printf ("\n Improper date entered");
return (0);
}
else
return (1);
}

```

Q. 6. (b) What are difference file opening modes in 'C'. Suppose a file contains student's records with each record containing name and age of a student. Write a 'C' program to read these records and display them in sorted order by name.

Ans. Different file opening modes are :

1. **"w" (write) :** If the file doesn't exist then this mode creates a new file for writing, and if the file already exists then the previous data is erased and the new data entered is written to the file.

2. **"a" (append) :** If the file doesn't exist then this mode creates a new file, and if the file already exists then the new data entered is appended at the end of existing data. In this mode, the data existing in the file is not erased as in "w" mode.

3. **"r" (read) :** This mode is used for opening an existing file for reading purpose only. The file to be opened must exist and the previous data of file is not erased.

4. **"wt" (write + read) :** This mode is same as "w" mode but in this mode we can also read and modify the data. If the file doesn't exist then a new file is created and if the file exists then previous data is erased.

5. **"rt" (read + write) :** This mode is same as "r" mode but in this mode we can also write and modify existing data. The file to be opened

must exist and the previous data of file is not erased.

6. **"a+" (append + read) :** This mode is same as the "a" mode but in this mode we can also read the data stored in the file.

```

main ( )
{
struct record
{
char name [20];
int roll;
float marks;
}
student, temp, stu [50];
FILE *fp;
int i, j, k = 0;
fp = fopen ("stu . dat", "rb");
if (fp == NULL)
{
printf ("Error in opening file \n");
exit (1);
}
while (fread (& student, size of (student),
1, fp) == 1) stu [k++] = student;
for (i = 0; i < k; i ++);
{
for (j = 0; j < k - 1 - i; j ++);
{
if (strcmp (stu [j] . name, stu [j + 1] .
name) > 0)
{
temp = stu [j];
stu [j] = stu [j + 1];
stu [j + 1] = temp;
}
}
}
printf ("\n NAME\t\t ROLL NO\t\t
MARKS\n\n");
for (i = 0; i < k; i++)
{

```

```

printf ("%s \t\t", stu [i] . name);
printf ("%d \t\t", stu [i] . roll);
printf (" %f \n", stu [i] . marks);
|
fclose (fp);
|

```

Q. 7. Attempt any one part :

Q. 7. (a) Write a 'C' program to convert a given decimal number to binary number using stack.

```

Ans. main ( )
|
int n1, bi, p, dg;
printf ("\n Enter any number");
scanf ("%d", & n1);
p = bi = 0;
while (n1 > 0)
|
dg = n1 % 2;
n1 = n1/2;
bi = bi + dg * pow (10, p);
p++;
|
printf ("The binary equivalent of %d is
%d", n1, bi);
getch ( );
|

```

Q. 7. (b) Write short notes on the following :

(i) Dynamic Memory Allocation In 'C'.

Ans. Dynamic Memory Allocation in 'C'

The memory allocation at the time of execution is called dynamic memory allocation. The allocation and release of the memory space can be done with the help of some built in functions whose prototypes are found in alloc.h and stdlib. h header files. These functions take memory from a memory area called heap and release this memory whenever not required, so that it can be used again for some other purpose.

Pointers play an important role in dynamic memory allocation because we can access the dynamically allocated memory only through pointers.

- (i) malloc () (ii) calloc ()
- (iii) realloc () (iv) free () .

Q. 7. (b) (ii) Creation of linked list in 'C'.

Ans. Creation of linked list In 'C'. A completely different way to represent a list is to make each item in the list part of a structure that also contains a "link" to the structure containing the next item. This type of list is called a linked list because it is a list whose order is given by links from one item to the next.

Each structure of the list is called a node and consists of two fields, one containing the item, and the other containing the address of the next item in the list. A linked list is therefore a collection of structures ordered not by their physical placement in memory but by logical links that are stored as part of the data in the structure itself. The link is in the form of a pointer to another structure of the same type. Such a structure is represented as follows :

```

struct node
|
int item;
struct node * next;
|;

```

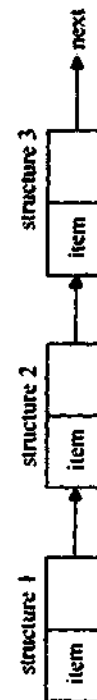


Fig. A linked list