

FIRST SEMESTER EXAMINATION, 2010-2011**COMPUTER CONCEPTS & PROGRAMMING IN C****Time : 3 Hours****Total Marks : 100**

Note : (1) Questions paper contains three sections.
(2) Attempt all questions.

SECTION - A

1. There are 10 multiple choice type of questions. Only one of the answers is correct. State the correct answer:

(10×1=10)

(a) Which of the following is not an operating system?

- (A) Linux (B) JAVA
(C) DOS (D) WINDOWS

Ans. (B) JAVA

(b) Program written in High Level Language is converted to machine code by

- (A) Operating System
(B) Assembler
(C) Compiler
(D) Machine Language

Ans. (C) Compiler

(c) Which of the following does not make use of conditions in its execution?

- (A) Loop (B) Case
(C) Function (D) Decision

Ans. (C) Function

(d) Which of the following is not a binary number?

- (A) 0 (B) 1
(C) 2 (D) 10

Ans. (C) 2

(e) Which of the following clause is used to include I/O function library?

- (A) `include#<io.h>`
(B) `include#<stdio.h>`
(C) `#include<stdio.h>`
(D) none of the above

Ans. (C) #include<stdio.h>

(f) Which of the following is a correct statement for checking a condition in the IF THEN ELSE statement?

- (A) `IF(A==B)`
(B) `IF(A = B)`
(C) `IF(A & B)`
(D) `IF(A *B)`

Ans. (A) `IF(A==B)`

(g) Which keyword is used to define structure?

- (A) `structure` (B) `STRUCTURE`
(C) `STRUCT` (D) none of the above

Ans. (C) `struct`

(h) Which method of input-output is used in stack?

- (A) First In First Out
(B) First In Last Out
(C) Both A & B
(D) None of the above

Ans. (B) First In Last Out

(i) How many elements will be there in A[5][5] array?

- (A) 5 (B) 25
(C) 50 (D) None of the above

Ans (B) 25

- (j) In a C program, the statement `for(i=0;i<5;i++)` will iterate the loop:
(A) 4 times (B) 5times
(C) 6 times (D) none of these

Ans. (C) 6 times

2. State whether the following statements are true or false (5 x 1 = 5)

- (a) An identifier in C must start with a letter or underscore. It is not allowed to have a space or a hyphen

Ans. (True)

- (b) Initialization of all elements of an array can be done at the time of declaration and definition

Ans. (True)

- (c) List is a non-linear data structure

Ans. (False)

- (d) A function in C cannot be invoked by another function

Ans. (False)

- (e) The name of an array is a pointer only to the first element, not the whole array

Ans. (True)

3. Fill in the blanks (5 x 1 = 5)

- (a) The function reads data from the keyboard

Ans. `scanf()`

- (b) An... .. is a sequence of operators and operands that reduces to a single value

Ans. expression

- (c) The statement will declare `pa` as a pointer to a character variable

Ans. `*char pa`

- (d) Mode opens existing file for read operations only

Ans. `read (r)`

- (e) is a pre defined macro of the C pre processor

Ans. `#`

SECTION-B

4. There are seven (07) parts. Attempt any five (05) parts.

- (a) (i) What are the difference in GUI and CUI interface? Give one example to explain your answer.

Ans.

Sl. no	GUI	CUI
1	It stands for command User Interface	It stands for Graphical User Interface.
2	It interact through the application by making uses of graphics	Interact with the application by making use of the codes.
3	It is allowed to work with any pointing device like mouse	It is allowed to work only with keyboard
4	It is based on graphic	It is based on text
5	The programs the user controls the flow of actions	The programmer controls the flow of actions
6	Example is Graphics	Example is DOS

(ii) Convert the following decimal number to equivalent binary number.

575, 311.55

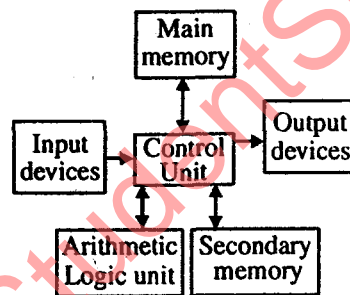
Ans. $575 = (1000111111)_2$

$311.55 = (100110111.100011001100....)_2$

- (b) (i) **Draw a neat schematic of digital computer and explain the role of each functional unit.**

Ans. A digital computer is used to store data in terms of digits and proceeds in discrete steps from one state to the next. The states of a digital computer typically involve binary digits which may take the form of the presence or absence of magnetic markers in storage medium, on-off switches. In digital computers, even letters, words and whole texts are represented digitally. A digital computer can only approximate a continuum by assigning large numbers of digits to a state description and by proceeding in arbitrarily small steps.

A digital computer consists of the following main components: Input devices, Memory, Central Processing Unit (CPU) and Output devices. The brain of computer is Central Processing Unit (CPU). A CPU has mainly three components: Main memory, Arithmetic Logic Unit (ALU) and Control Unit.



A Block Diagram of a Digital Computer

The main memory is used to hold the data being processed and holds the instruction for doing the processing.

ALU is used to perform only arithmetic and logic operations. The arithmetic operations are addition, subtraction, multiplication and division and logic operation are OR, AND and NOT.

Control unit is used to monitor and control all the operations of the computer. It also acts as an interface between main memory and ALU and main memory and secondary memory.

- (ii) **what is algorithm? Discuss the basic characteristic of an algorithm**

Ans. Algorithm is a list of well-defined instructions for completing a task. Starting from an initial state, the instructions describe a computation that proceeds through a well-defined series of successive states, eventually terminating in a final ending state. The transition from one state to the next is not necessarily deterministic; some algorithms, known as randomized algorithms, incorporate randomness.

Characteristics of an algorithm

- (i) Algorithm should be finite.
- (ii) Each instruction should be precise and unambiguous
- (iii) Algorithm should end
- (iv) It should give result

- (c) (i) **Draw a flow chart for arranging three numbers a, b, c in ascending order.**

- (ii) **What is role of switch statement in C programming language? Give the syntax of SWITCH statement' with suitable example.**

Ans. (ii) The switch statement is another form of the multi way decision statements. It is well structured, but can only be used in certain cases where;

- Only one variable is tested, all branches must depend on the value of that variable. The variable must be an integral type. (int, long, short or char).
- Each possible value of the variable can control a single branch. A final, catch all, default branch may optionally be used to trap all unspecified cases.

It is very similar to if statement but can not replace in all cases.

Example: Write a program for simple calculator using switch statement.

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

```
main()
```

```
{
```

```
int menu, numb 1, numb2, result;
```

```
printf("enter two numbers :A");
```

```
scanf("%d%d",&numb 1, &numb2);
```

```
printf("enter in choice\n");
```

```
printf("1 =addition\n");
```

```
printf("2=subtraction\n");
```

```
printf("3=multiplication\n");
```

```
printf("4=division\n");
```

```
scanf("%d", &menu);
```

```
switch( menu)
```

```
{
```

```
case 1: result = numb 1 + numb2; break;
```

```
case 2: result = numb 1 - numb2; break;
```

```
case 3: result = numb 1 * numb2; break;
```

```
case 4: result = numb1 / numb2; break;
```

```
default: printf("Invalid option selected\n");
```

```
}
```

```
if( menu == 1)
```

```
printf("%d plus %d is %d\n", numb 1, numb2,
```

```
result); else if( menu == 2 )
```

```
printf("%d minus %d is %d\n", numb1, numb2,
```

```
result);
```

```
else if( menu == 3 );
```

```
printf("%d multiplication %d is %d\n",numb 1
```

```
,numb2,result); else if( menu == 4 )
```

```
printf("%d division %d is %d\n",numb 1, numb2,
```

```
result); else
```

```
printf("Invalid option selected\n");
```

```
}
```

(d) (i) **What are functions? What is the advantage off using multiple functions in a program?**

Ans. Function in C language is a self defined block of code and contain set of statements which performs a specific task. It has a name and it is reusable i.e. it can be executed from as

many different parts in a C Program as required. It also optionally returns a value to the calling program So function in a C program has some properties discussed below.

- Every function has a unique name. This name is used to call function from "main()" function. A function can be called from within another function.
- A function is independent and it can perform its task without intervention from or interfering with other parts of the program.
- A function performs a specific task. A task is a distinct job that your program must perform as a part of its overall operation, such as adding two or more integer, sorting an array into numerical order, or calculating a cube root etc.
- A function returns a value to the calling program. This is optional and depends upon the task your function is going to accomplish.

Advantage of using multiple functions in program:--

- Debugging is easier
- It is easier to understand the logic involved in the program
- Testing is easier
- Recursive call is possible
- Irrelevant details in the user point of view are hidden in functions
- Functions are helpful in generalizing the program

(d) (ii) **What do you mean by call by value technique for function call? Explain with a suitable example.**

Ans. In the call by value you can use the concept of actual and formal parameter. It means the value of actual parameters is passed and they are copied into the formal parameters. Since the value of actual and formal parameter is stored in different memory location so the changes in formal parameter do not change the value of actual parameters. For example,

```

#include < stdio.h>
#include < conio.h>
int compute (int, int);
main( )
{
    int number1 = 100, number2= 500;
    printf ("The value before calling the function");
    printf("number1 = %d and number2 =%d \n",
    number1 , number2); comput(number1, number2);
    printf("The value after calling the function");
    printf("number1 = %d and number2 =%d \n",
    number1, number2);
}
int compute(int num 1, int num2)
{
    num1= num1 + 100;
    num2= num2 + 100;
}

```

- (e) (i) What is difference in searching and sorting? Write an algorithm to sort a list containing ten numbers

Ans. Sorting is a process of arranging the data in a particular order. The order may be ascending or descending order for the numeric values and dictionary order for the alphanumeric values

- (e) (ii) what are the difference in array and structure? Explain with the help of example.

Ans.

Sl. No.	Array	Structure
1.	It is used for same data type	It is used for different data type
2.	It is a static memory allocation. It uses the subscript to access the array elements.	It is a dynamic memory allocation. It uses the dot(.) Operator to access the structure members.
3.	It has a base pointer to point particular memory location	It is not a pointer
4.	An array can't have bit fields.	Structure can contain bit fields
5.	Example: char number [10];	Example: Struct month <pre> { int num _days; char name[10]; }; </pre>

whereas Searching is process to find whether a data is present in the set of data or not.

There are many searching and sorting algorithms available. Some are faster than others that depend upon the behavior of the elements. An algorithm of bubble sort is :

Step 1 : Compare the first and second data items. If the first data item is greater than the second, then make an interchange.

Step 2 : Compare second and third data items. If the second one is greater than the third one then make an interchange.

Step 3 : The process is repeated till the last data item is reached.

Step 4 : When the last data item is reached, it is said to be one pass. At the end of the first pass, the largest data item is bubbled out. That is it occupies the last position.

Step 5 : The steps 1 to 4 is repeated for the data items between] to 9, because the 10th data item is already sorted. At the end of second pass, the next highest data item occupies its right place.

Step 6 : The steps are repeated till the last pass 9 is reached.

Step 7 : At the end of the last pass, the entire list sorted.

- (f) (i) **What is a pointer? How pointers are declared in C? Illustrate with a suitable example.**

Ans. Pointer: A pointer is a variable that contains the memory location of another variable. The syntax is as shown below. You start by specifying the type of data stored in the location identified by the pointer. The asterisk tells the compiler that you are creating a pointer variable. Finally you give the name of the variable.

Data_type *variable name

Example:

```
int *ptr; float *string;
```

Address operator: Once we declare a pointer variable we must point it to something we can do this by assigning to the pointer the address of the variable you want to point as in the following example:

```
ptr = &num;
```

This places the address where num is stored into the variable ptr. If num is stored in memory 21260 addresses then the variable ptr has the value 21260.

Example:

```
main ()
{
    int *ptr; int sum;
    sum = 45;
    ptr = &sum;
    printf("\n Sum is %d\n", sum);
    printf("\n The sum pointer is %d", ptr);
}
```

- (g) (i) **What are the difference in stack and list? Explain how an element can be deleted from stack and list.**

Ans.

Sl. No.	Linked list	Stack
1.	It is a linearly arranged collection of elements that allows insertion and deletion at any place in the sequence.	It is last In First Out (LIFO) That is insertion and deletion always allows from same end points.
2.	It is always implemented by dynamic memory	It is implemented by dynamic as well as static memory both

We will get the same result by assigning the address of num to a regular (non pointer) variable. The benefit is that we can also refer to the pointer variable as *ptr the asterisk tells to the computer that we are not interested in the value 21260 but in the value stored in that memory location. While the value of pointer is 21260 the value of sum is 45 however we can assign a value to the pointer * ptr as in *ptr=45.

- (f) (ii) **Write a program segment in c to swap the value of two variable using pointers**

Ans.

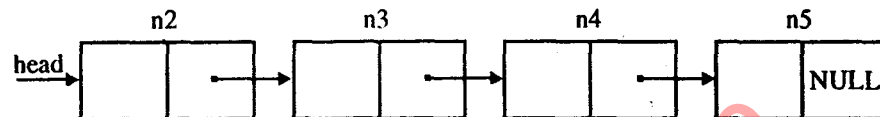
```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a;
    int b;
    printf("Enter the value of a and b :");
    printf("%d%d",&a,&b );
    swap( &a, &b)
    printf("The value of a and b after swap is %d and %d", a, b); getch();
}

void swap(int *px, int *py)
{
    int temp;
    temp = *px;
    *px = *py;
    *py = temp;
}
```

Deletion from linked list : A deletion operation is used to delete a node from a linked list. A node may be deleted from a linked list in the following possible conditions. But here performed only deletion from the specified position in a linked list.

- From the beginning of a linked list
- From the end of a linked list
- From the specified position in a linked list

Consider a linked list of six nodes and suppose the six nodes are n2, n3, n4 and n5.



To delete a node from a linked list after node n2. The resultant linked list looks like as follows by the following statements

n2 -> next = n4;

free (n3);



Deletion from Stacks : It is called POP

```

int pop (struct stack s)
{
    int x;
    if(s.top == -1)
        printf("Stack is empty.\n"); else
    {
        x = s.items[s.top];
        s.top = s.top - 1;
        return x;
    }
}
  
```

(g) (ii) What do you mean by C processor? Give example of any two processor directives.

Ans. The C preprocessor (cpp) is the preprocessor for the C programming language. In many C implementations, it is a separate program invoked by the compiler as the first part of translation. The preprocessor handles directives for source file inclusion (#include), macro definitions (#define), and conditional inclusion (#if). The language of preprocessor directives is agnostic to the grammar of C, so the C preprocessor can also be used independently to process other types of files.

Two processor directives.

#line (directive)

Causes the compiler to think that the line number of the next source line is given by <constant>, and the current input file is given by <identifier>.

Syntax: #line <constant> [<identifier>]

If <identifier> is missing, the current file name remains unchanged.

Example:

```
#line 55 main.cpp
```

#error (directive)

This is used to issues an error message and the Syntax is defined as follows

```
#error <message>
```

If this line of code is compiled by the compiler, a fatal error message will be issued for this line and include the text defined by <message>.

Example: -

```
#if !defined(MODEL)
```

```
#error Building model not defined
```

```
#endif
```

SECTION - C

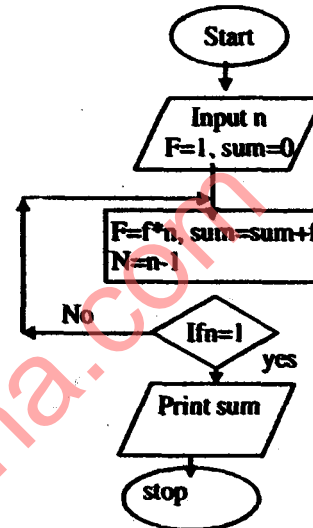
5. This section contains seven programming problems Attempt any five problems:-

- (a) Draw a flowchart and write a function in C to calculate factorial of given number and also write a program to calculate the sum of following series using above function:-
 $S = 1! + 2! + 3! + \dots + n!$

Ans.

```
#include<stdio.h>
#include<conio.h>
void factorial(void)
void main()
{
    int i,num,f;
    int sum=0;
    printf("enter the number");
    scanf("%d",&num);
    for(i=1;i<=num;i++)
        sum = sum + factorial(i);
    printf("sum is %d",sum);
    getch();
}
int factorial(int x)
{
```

```
if(x== 1)
    return 1;
else
    return x * factorial( x - 1);
}
```



- (b) Write a program in C that accepts roll number and name of student of a class size of 100 students along with the marks obtained by them in physics, chemistry and mathematics. Print roll number and name of the top ten students in the order of merit. The merit is based on the sum of marks obtained in three subjects.

Ans.

```
#include<stdio.h>
#include<conio.h>
struct student
{
    char name [15];
    int rollno;
    int phy;
    int che;
    int math;
    void 3; main()
};
```



```

struct student * str, * stud[ 1 00];
printf("enter the values of student record");
for(i=0;i<100;i++)
{ printf("enter the record %d",i);
scanf("%s%d%d%d%d",stud[i]->name, &stud[i]-
>rollno, &stud[i]->phy, &stud[i]->che,
&stud[i]->math);
}
for(i=0;i<100;i++)
{ printf(" records are");
printf("%s%d%d%d%d",stud[i]-> name, stud[i]-
>rollno, stud[i]->phy, stud[i]->che, stud[i]->math);
}
for(i=0;i<100;i++)
{ if(stud[i]->phy+stud[i]->che+stud[i]->math <
stud[i+ 1 ]->phy+stud[i+ 1 ]->che+stud[i+ 1 ]-
>math)
str[i]->name=str[i+ 1 ]->name;
str[i]->rollno=str[i+ 1 ]->rollno;
str[i]->phy=str[i+ 1 ]->phy;
str[i]->che=str[i+ 1 ]->che;
str[i]->math=str[i+ 1 ]->math;
}
printf(sorted records are");
for(i=0;i<100;i++)
{printf("%s%d%d%d%d",stud[i]-> name, stud[i]-
>rollno, stud[i]->phy, stud[i]->che, stud[i]->math);
}
getch();
}

```

- (c) Write a program in C that accepts the length and width of a rectangle and print the area. The area of a rectangle is calculated by a function and returns its value to the main program where it is printed. The values of length and width of rectangle are accepted from the keyboard. Also give the flowchart and algorithm

Ans.

```

#include<stdio.h>
float call(int, int);
void main()
{

```

```

int length, width;
float area;
printf("Enter values of length and width");
scanf("%d %d", & length, &width);
area = call(length,width); printf("Area= %f", area);

```

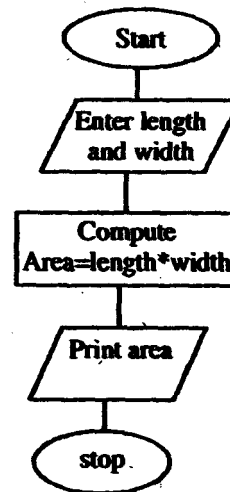
```

}
float call(int l, int w)
{
return(l*w);
}

```

Algorithm

1. Start
2. Enter the values of length and width of the rectangle
3. Calculate area = length * width
4. Print area
5. Stop



5. (d) Write a program in C to calculate the sum of the following series upto nth term

$$F(x) = X^1 - X^3 + X^5 - X^7 + \dots$$

Ans.

```

#include<stdio.h>
#include<math.h>

```

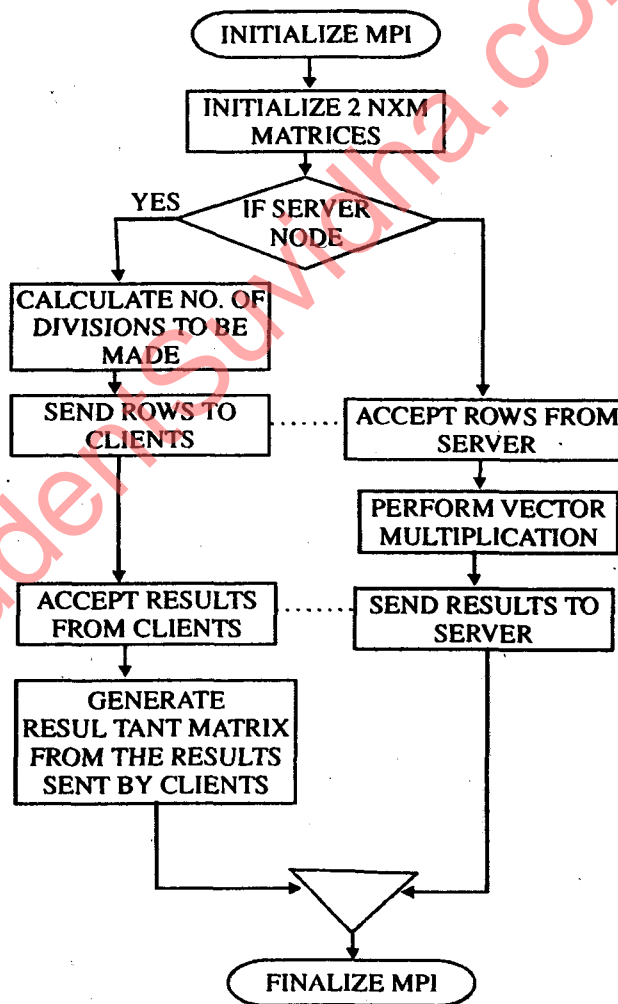
```

void main()
{
    int x, i, n, count = 1;
    long f = 0;
    printf("Enter the value of x");
    scanf("%d" &x);
    printf("Enter the value of limit n");
    scanf("%d" &n);
    for(i=1 ;i<=n;i=i+ 2)
    {
        if(count % 2 == 0)
            f= f - pow(x,i);
        else
            f= f + pow(x,i);
        count = count + 1;
    }
    printf("Sum of series = %ld", t);
}

```

(e) Draw flow chart & write a program in C to print the multiplication of two Matrices A & B of size N*N

Ans.



Program:

```

#include<stdio.h>
#include<conio.h>
void main()
{
int k,o,ij,m1 [10][10],m2[10][10],m- mul[10][10];
clrscr();
printf("enter order for matrix\n");
scanf("%d",&o);
printf("\nEnter %d numbers for 1st%d*%d
matrix\n",o*o,o,o);
for(i=0;i<=o-1;i++)
{
for(j=0;j<=o-1;j++)
{
scanf("%d",&m1[i][j]);
}
}
printf("Enter %d numbers for 2nd %d*%d
matrix\n",o*o,o,o);
for(i=0;i<=o-1;i++)
{
for(j=0;j<=o-1;j++)
{
scanf("%d",&m2[i][j]);
}
}
printf("\nPrinting 1ST Matrix\n");
for(i=0;i<=o-1;i++)
{
printf("[");
for(j=0;j<=o-1;j++)
{
printf("%d",&m1[i][j]);
}
printf("]\n");
}
//Multiplying matrix
for(i=0;i<=o-1;i++)
{
for(j=0;j<=o-1;j++)
{
m_mul[i][j]=0;
for(k=0;k<=o-1;k++)

```

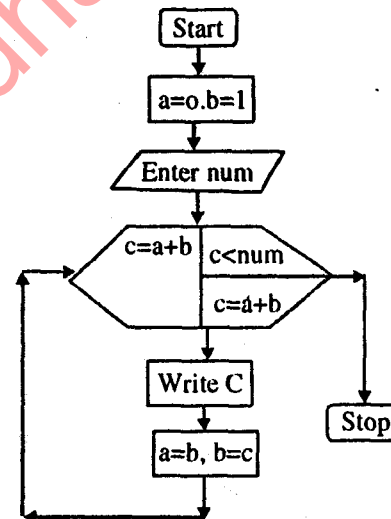
```

{
m_mul[i][j]=m1[i][k]*m2[k][j]+m_mul[i][j];
}
}
printf("\n\nMultiplication of Two Matrix\n\n");
for(i=0;i<=o;++)
{
printf("[");
for(j=0;j<=o-lj++)
{
printf("%d",m_mul[i][j]);
}
printf("]\n");
} getch();
}

```

(f) Draw Flowchart & write a program in C to calculate the sum of Fibonacci series upto 100 terms.

Ans. FLOW CHART:

**Program:**

```

#include<stdio.h>
#include<conio.h>
void main()
{
int a=0,b=1,c,num;// a=0 & b=lare
default values

```

```

clrscr();
printf("Program to print Fibonacci
series\n");
printf("Enter a number till Fibonacci
series should be printed\n");
scanf("%d",&num);
printf("\n%d\n%d\n",a,b);
for( c=a+b;c<=num;c=a+b)
{
    printf("%d\n",c );
    a=b;
    b=c;
}
getch();
}

```

(g) Using dynamic memory allocation, write a program in C to accept element of a matrix of size 3*3 & print transpose of it.

Ans.

```

void main ()
{
    int **a, x;
    a = malloc(sizeof(int *) * 3);
    for(x = 0; x < 10; X ++ )
    {a[x] = malloc(sizeof(int) * 3);
    printf("enter number of rows & columns of
matrix maximum 10 elements\n");
    scanf("%d%d", &r 1 ,&c 1);
    printf("row wise\n");
    for(i=0;i<a;i++)
    {
        for(j=0;j<a;j++)
        scanf("%d",&A[i][j]);
    }
    printf("the transpose of the matrix is");
    for(i=1;i<=a;i++)

```

```

{
    for(j= 1 ;j<=j++;)
    {
        B[ij]=A[j,i];
    }
}
for(i=1 ;i<=a;i++)
{
    for(j=1 ;j<=a;j++)
    {
        printf("%d",B[i] [j]);
    }
}
}

```

