

EXP NO : 1a

PRINT NAME AND ADDRESS

DATE:

AIM:

To write a C program to print name and address

REQUIREMENTS:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM:

1. Start the program.
2. Print the name and address
5. Stop the program.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr(); // clears the output window
    printf("\n Name : David"); // prints the name
    printf("\n Address : London"); // prints the address
    getch();
}
```

OUTPUT

RESULT

Thus a c program to print name and address was written, compiled, executed and output was verified.

EXP NO : 1b.

SIMPLE AND COMPOUND INTEREST

DATE:

AIM:

To write a C program to calculate simple and compound interest.

REQUIREMENTS:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM:

1. Start the program.
2. Get the values for P, N and R from the user as input.
3. Using the formula calculate simple and compound interest.

• $si = p * n * r / 100;$

• $ci = p * \text{pow}((1 + r / 100), n) - p$

4. Print the values for simple and compound interest.

5. Stop the program.

PROGRAM

```
#include<stdio.h>
void main()
{
float p,n,r,si,ci;
clrscr();
printf("Enter the values for P,N,R");
scanf("%f%f%f",&p,&n,&r);
si=p*n*r/100;
ci=p*pow((1+r/100),n)-p;
printf("Simple Interest = %f",si);
printf("\nCompound Interest = %f",ci);getch();
}
```

OUTPUT

RESULT:

Thus, the program to calculate simple and compound interest was written, compiled, executed and output was verified.

EXP NO : 2(i)

SWAPPING TWO VARIABLES

DATE:

AIM:

To write a C program to calculate swap two numbers

i) using third variable

ii) without using third variable

REQUIREMENTS:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM: (using third variable)

1. Start the program.
2. get two values x and y from user as input.
3. print the values before swapping
4. swap the values by using third variable.
5. Stop the program.

ALGORITHM: (without using third variable)

1. Start the program.
2. get two values x and y from user as input.
3. print the values before swapping
4. swap the values by using third variable.
5. Stop the program.

PROGRAM(using third variable)

```
#include <stdio.h>

int main()
{
int x, y, temp;
clrscr();

printf("Enter the value of x and y\n");
scanf("%d%d", &x, &y);
printf("Before Swapping\nx = %d\ny = %d\n",x,y);

temp = x;
x = y;
```

```
y = temp;
printf("After Swapping\nx = %d\ny = %d\n",x,y);
getch();
return 0;
}
```

OUTPUT

MLPTC

PROGRAM(without using third variable)

```
#include <stdio.h>
int main()
{
int a, b;
clrscr(); printf("Enter two integers to swap\n");
```

```
scanf("%d%d", &a, &b);  
printf("Before Swapping\na = %d\nb = %d\n",a,b);  
a = a + b;  
b = a - b;  
a = a - b;  
printf("After swapping \na = %d\nb = %d\n",a,b);  
getch();  
return 0;  
}
```

OUTPUT

RESULT

Thus a c program to swap two values i) using third variable and ii) without third variable was written, compiled, executed and output was verified.

EXP NO : 3 CONVERRT NUMBER OF DAYS INTO MONTHS AND DAYS

DATE:

AIM:

To write a c program to convert a given number of days into months and days using integer arithmetic operators.

REQUIREMENT:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM:

1. Start the program.
2. Get days from user as input.
3. Calculate months and days using calculation $\text{months}=\text{days}/30$ and $\text{days}=\text{days}\%30$;
4. Print Months and Days
5. Stop the program.

PROGRAM

```
#include<stdio.h>
void main()
{
    int months,days;
    printf("Enter days\n");
    scanf("%d",&days);
    months=days/30;
    days=days%30;
    printf("Months=%d Days =%d",months,days);
}
```

OUTPUT

RESULT

Thus a c program to convert a given number of days into months and days was written, compiled, executed and output was verified.

EXP NO : 4 EVALUATION OF VARIABLE EXPRESSION

DATE:

AIM:

To write a c program use of variables in expression and their evaluation

REQUIREMENT:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM:

1. Start the program.
2. Get three values a,b,c from user as input.
3. Assign values to variables & evaluate expression

4. Print the variable
7. Stop the program.

PROGRAM

```
#include<stdio.h>

void main()
{
    float a,b,c,x;
    a=9,b=12,c=3;
    x=a-b/3+c*2-1;
    printf("x=%f\n",x);
}
```

Output

RESULT

Thus a c program to use of variables in expression and evaluation was written, compiled, executed and output was verified.

EXNO:5 FAHRENHEIT TO CELSIUS

DATE:

AIM:

To write a c program to converts the given temperature in Fahrenheit to Celsius using preprocessor.

REQUIREMENT:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM:

ALGORITHM

1. Start the program.
2. Write a macro name f and define value for it.
3. Calculate Fahrenheit to Celsius using formula

4. Print the values
5. Stop the program.

PROGRAM

```
#include<stdio.h>
#define f 40
int main()
{
    float c;
    c=(f-32)/1.8;
    printf("Fahrenheit to Celsius:%f ",c);
    return(0);
}
```

Output

RESULT

Thus the program temperature in Fahrenheit to Celsius using preprocessor was written, compiled, executed and output was verified.

EXP NO : 6

LARGEST OF GIVEN THREE NUMBERS

DATE:

AIM:

To write a c program to find the largest of the given three numbers.

REQUIREMENT:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM:

1. Start the program.
2. Get three values a,b,c from user as input.
3. Using if statement check whether a is greater than b if true then check whether a is greater than c.
4. If true,print “a is largest.”else print “c is largest”.
5. If the if statement becomes false,then check whether b is greater than c.
6. If the condition is true print “b is largest”.
7. Stop the program.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c;
    clrscr();
    printf("\n\nEnter The value of a,b,c\n\n");
    scanf("%d%d%d",&a,&b,&c);
    if(a>b)
    {
        if(a>c)
            printf("largest=%d",a);
        else
            printf("largest=%d",c);
    }
    else
    {
        if (b>c)
            printf("largest=%d",b);
        else
            printf("largest=%d",c);
    }
    getch();
}
```

OUTPUT

RESULT

Thus a c program to find the largest of three numbers was written, compiled, executed and output was verified.

EXP NO : 7a. FACTORIAL OF A GIVEN NUMBER

DATE:

AIM:

To write a C program to find the factorial of a given number

.REQUIREMENTS:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM

1. Start the program.
2. Declare the necessary variables.
3. Get the number from user to find the factorial.
4. Using for loop find the factorial of the given number.
5. Print the factorial of the given number.
6. Stop the program.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
void main()
```

```
{  
int i,n,f;  
clrscr();  
printf("Enter the number to find the factorial\n");  
scanf("%d",&n);  
f=1;  
for(i=1;i<=n;i++)  
f=f*i;  
printf("The factorial of a number %d is %d",n,f);  
getch();  
}
```

OUTPUT

RESULT:

Thus, the program to calculate the factorial of a given was written, compiled, executed and output was verified.

EXP NO : 7b.

PRIME NUMBERS

DATE:

AIM:

To write a C program to print prime numbers up N times.

REQUIREMENTS:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM

1. Start the program.
2. Declare the necessary variables.
3. Get the values from user
4. Using for loop and if statement to check whether it is a prime or not
5. Print the prime number
6. Stop the program.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int N, i, j, isPrime, n;
    printf("To print all prime numbers between 1 to N\n");
    printf("Enter the value of N\n");
    scanf("%d",&N);
    /* For every number between 2 to N, check whether it is prime number or not */
    printf("Prime numbers between %d to %d\n", 1, N);

    for(i = 2; i <= N; i++){
        isPrime = 0;
        /* Check whether i is prime or not */
```

```

    for(j = 2; j <= i/2; j++){
/* Check If any number between 2 to i/2 divides I completely If yes i cannot be prime no */
        if(i % j == 0){
            isPrime = 1;
            break;
        }
    }

    if(isPrime==0 && N!= 1)
        printf("%d ",i);
    }
    getch();
    return 0;
}

```

OUTPUT

RESULT:

Thus, the program printing prime numbers was written, compiled, executed and output was verified.

EXP NO : 8 TOTAL MARKS USING ARRAY OF STRUCTURES

DATE:

AIM

To write a c program to calculate total marks using array of structures.

REQUIREMENT:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM

1. Start the program.
2. Declare a structure student with variables name, roll no, mark, tot.
3. Declare the necessary variables.
4. Create a structure variable using arrays.
5. Get the student roll no, name, mark using for loop.
6. Calculate the total marks using arithmetic operator.
7. Using for loop display name, roll no and total for each student.
8. Stop the program.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
struct stu
{
int no;
char stuname[100];
int mark[20];
int totalmarks;
};
void main()
{
clrscr();
struct stu s[50];
int n,i,j;
printf("Enter Number of Students : ");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("\n Enter Number,Student name and 6 Marks one by one\n");
scanf("%d%s",&s[i].no,s[i].stuname);
for(j=0;j<=5;j++)
{
```

```

scanf("%d",&s[i].mark[j]);
}
}
printf("\n*****");

printf("\nNumber\tName\tTotalMarks");
printf("\n*****\n");
for(i=0;i<n;i++)
{
s[i].totalmarks=0;
for(j=0;j<6;j++)
{
s[i].totalmarks=s[i].totalmarks+s[i].mark[j];
}
printf("%d\t%s\t%d\n",s[i].no,s[i].stuname,s[i].totalmarks);
}
getch();
}

```

OUTPUT

RESULT

Thus a c program to display student total marks was written, compiled, executed and output was verified.

EXP NO : 9 IMPLEMENTATION OF POWER FUNCTION

DATE:

AIM

To write a c program using the function power(a,b) to calculate the value of a raised to b.

REQUIREMENT:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM

ALGORITHM

1. Start the program.
2. Declare the necessary variables.
3. Get three values from user as input.
4. Declare a function named power.
5. Define the function outside the main () function.
6. Calculate the function power(a,b)
7. Print the value
8. Stop the program.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
int power(int a,int b);
int main()
{
int a,b,value;
printf("enter the value of a and b: ");
scanf("%d%d",&a,&b);
value=power(a,b);
printf("the value of a = %d raised to b = %d is %d",a,b,value);
getch();
return 0;
}
int power(int a,int b)
{
int i,value;
value=a;
for(i=1;i<b;i++)
{
value=value*a;
}
return value;
}
```

OUTPUT

RESULT

Thus a c program using the function power(a,b) to calculate the value of a raised to b was written, compiled, executed and output was verified.

EXP NO : 10

LENGTH OF STRING USING POINTERS

DATE:

AIM

To write a c program to find the length of the string using pointers.

REQUIREMENT:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM

1. Start the program.
2. Declare the needed pointer variables.
3. Get the string from user using gets() function.
4. using while find length by incrementing pointer variables.
5. Finally print the length of the string.
6. Stop the program.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
int l=0;
char *p;
char name[20];
p=name;
clrscr();
printf("\n enter a string\n");
gets(name);
while(*p !='\0')
{
l++;
```

```
p++;  
}  
printf("\n");  
printf("the length of the string is %d\t",l);getch();  
}
```

OUTPUT

RESULT

Thus a c program to find the length of the string using pointers was written, compiled, executed and output was verified.

EXP NO : 11

SUM OF ALL INDIVIDUAL DIGITS AND PRINT IN

DATE:

REVERSE

AIM:

To write a C program to calculate Sum of all individual digits and also print the above number in reverse order.

REQUIREMENTS:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM:

1. Start the program.
2. Declare the variables count, sum, reverse.
3. Initialize the count and sum to zero.
4. Get an integer number from user as input.
5. Assign m with n.
6. Using while statement, find the sum of digits and number of digits using divide and modulo operation.
7. Print the sum of digits and number of digits of the given number.
8. Print the numbers in reverse order.
9. Stop the program.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
void main()
{
int count=0,sum=0,rev;
int r,n,m;
clrscr();
printf("Enter an integer number");
scanf("%d",&n);
m=n;
printf("\n Reverse order of %d is",n);while(m!=0)
{
count=count+1;
r=m%10;
```



```
printf("%d",r);
sum=sum+r;
m=m/10;
}
printf("\n Sum of digit for %d is %d",n,sum);
printf("\n");
printf("\n Number of digits in %d is %d",n,count);
getch();
}
```

OUTPUT

RESULT:

Thus, the program to calculate Sum of all individual digits and also print the above number in reverse order was written, compiled, executed and output was verified.

EXP NO : 12a

FIBONACCI SERIES

DATE:

AIM:

To write a C program to print Fibonacci series upto N terms.

REQUIREMENTS:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM:

1. Start the program.
2. Declare the variables f1,f2,f3,n,i,sum
3. Initialize the sum to zero.
4. Get an input from user
5. Using for loop, calculate Fibonacci numbers and its sum.
6. Print the Values
7. Stop the program.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
void main()
{
int f1,f2,f3,n,i,sum=0;
f1=-1,f2=1;
printf("Enter the total number \n");
scanf("%d",&n);
for(i=0;i<n;i++)
{
f3=f1+f2;
f1=f2;
f2=f3;
sum=sum+f3;
printf("%d\n",f3);
}
printf("the sum is %d",sum);
getch();
}
```

OUTPUT

RESULT

Thus, the program to calculate Fibonacci series up to N terms and its sum was written, compiled, executed and output was verified.

EXP NO : 12b

LEAP YEAR

DATE:

AIM:

To write a C program to print whether a given year is leap or not.

REQUIREMENTS:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM:

1. Start the program.
2. Get an input from user
3. Calculate leap year.

4. Print the Values
5. Stop the program.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
void main()
{
int year,
printf("enter a year");
scanf("%d",&year);
if(year%400==0)
printf("%d is leap year\n",year);
elseif (year%100==0)
printf("%d is not a leap year\n",year);
elseif(year%4==0)
printf("%d is leap year\n",year);
else
printf("%d is not a leap year\n",year);
}
```

OUTPUT

RESULT

Thus, the program to print whether a year or not was written, compiled, executed and output was verified.

EXP NO : 13

PRINTING STRING ARGUMENTS IN REVERSE

DATE:

AIM:

To write a C program to print the string arguments in reverse order using command line arguments.

.REQUIREMENTS:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM:

1. Start the program.
2. Declare an array to store the strings.
3. Give the input using command line arguments.
4. Using for loop reverse the string using strcat function.
5. Get the length of the string using strlen () function.
6. Display the string in reverse using for loop.

7. Stop the program.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main(int argc, char*argv[])
{
clrscr();
char str[50];
for(int i=1;i<=argc;i++){
strcat(str,argv[i]);
}
int l=strlen(str);
for(i=l-1;i>=0;i--)
printf("%c",str[i]);
getch();
}
```

OUTPUT

RESULT:

Thus, the program to print the string arguments in reverse using command line arguments was written, compiled, executed and output was verified.

EXP NO : 14 ARRANGE NAMES IN ALPHABETICAL ORDER

DATE:

AIM:

To write a C program to arrange the given N names in alphabetical order.

REQUIREMENTS:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM:1. Start the program.

2. Declare the needed variables.

3. Get the number of strings from user.

4. Get the strings from user as input.

5. Using strcmp() function compare the first two strings.

6. Swap the greater with the smaller and continue for next strings.

7. Print the arranged strings.

8. Stop the program.

PROGRAM

```
#include<string.h>
```

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

```

void main()
{
char name[20][30],temp[30];
int n,v,i,j;
clrscr();
printf("Give the number of names to sort\n");
scanf("%d",&n);
printf("Give the names one by one\n");
for(i=0;i<=n;i++)
gets(name[i]);
for(i=0;i<n;i++)
for(j=i+1;j<=n;j++)
{
v=strcmp(name[i],name[j]);
if(v>0)
{
strcpy(temp,name[i]);
strcpy(name[i],name[j]);
strcpy(name[j],temp);
}
}
printf("The sorted names are \n"); for(i=0;i<=n;i++)
puts(name[i]);
getch();
}

```

OUTPUT

MLPTC

RESULT:

Thus, the program to arrange the given N names in alphabetical order was written, compiled, executed and output was verified.

EXP NO : 15

COUNT NUMBERS AND CHAR IN STRING

DATE:

AIM:

To write a C program to count the numbers and chars in string.

REQUIREMENTS:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM:

1. Start the program.
2. initialize variables, Get the string value as input.
3. Print the character and numbers in string
5. Stop the program.

PROGRAM

```
#include<stdio.h>

int main()
{
    char line[150];
    int i,c,d;
    c=d=0;
    printf("Enter a line of string:\n");
    gets(line);
    for(i=0;line[i]!='\0';++i)
    {
```

```
if((line[i]>='a'&& line[i]<='z') || (line[i]>='A'&& line[i]<='Z'))
    ++c;
else if(line[i]>='0'&&c<='9')
    ++d;
}
printf("\nCharacters: %d",c);
printf("\nDigits: %d",d);
return 0;
}
```

Output

RESULT:

Thus, the program to count the number of characters and numbers in string was written, compiled, executed and output was verified.

EXP NO : 16

SORTING ARRAYS USING FUNCTION

DATE:

AIM:

To write a C program to sort an array of integers using function.

REQUIREMENTS:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBO C

ALGORITHM:

1. Start the program.
2. Declare the array and variables
3. Get the values from the user as input
4. Declare a function named sortarray.
5. Define the function outside the main() function.
6. In sortarray function take the first element and compare with the second element. If it is small don't interchange or else interchange the two elements.
7. Then Compare the second element with the third element. If it is small don't interchange or else interchange two elements.
8. This process is continued till (n-1)th element is compared with the nth element.
9. At the end of the first scan, the highest element comes to the nth position.
10. Repeat steps 3,4,5 till the element is comes to the sorted order.
11. Print the sorted array of integers.
12. Stop the program.

PROGRAM

```
#include<stdio.h>  
  
int sortArray(int);
```

```

int array[100];

void main()
{
int i,size;

printf("\n Program to sort array using Function \n\n");

printf("\n Enter the size of the array:");

scanf("%d",&size);

printf("\n Enter the %d elements of array: \n",size);

for(i=0;i<size;i++)
{
printf("\n array[%d]=",i);
scanf("%d",&array[i]);
}

sortArray(size);

printf("\n The sorted elements of array are :");

for(i=0;i<size; i++)
{
Printf("%d",array[i]);
}

sortArray(n)
{
int temp=0,i,j;
for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
{
if(array[i]>array[j])
{
temp=array[i];
array[i]=array[j];

```

```
array[j]=temp;  
}  
}  
}  
}
```

Output

RESULT:

Thus, the program to sort an array of integers using function was written, compiled, executed and output was verified.

DATE :

AIM :

To write a C program to calculate the subject wise and student wise total and store them as a part of the structure.

REQUIREMENTS :

- Computer with PENTIUM IV/DUAL CORE PROCESSORS
- TURBOC Compiler

ALGORITHM

- 1.Start the Program
- 2.Declare a structure student with variable name s1,s2,s3,s4
- 3.Declare the necessary variables
- 4.Create a structure variable using arrays
- 5.Get the student name,mark using for loop
- 6.Calculate the subject and student wise totals.
- 7.Using the for loop display name, s1,s2,s3,s4 and total for each student
- 8.Stop the Program.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
struct mark
{
int s1,s2,s3,s4;
char name[20];
int regno,total;
};
void main()
{
clrscr();
struct mark s[3];
struct mark subtotal={0};
```

```

printf("Give 3 students marks in 3 subjects \n");
printf("Enter the details line by line \n");
for(i=0;i<3;i++)
{
scanf("%d%s%d%d%d",&s[i].regno,s[i].name,&s[i].s1,&s[i].s2,&s[i].s3);
}
for(j=0;j<3;j++)
{
s[i].total=s[i].s1+s[i].s2+s[i].s3;
subtotal.s1=subtotal.s1+s[i].s1;
subtotal.s2=subtotal.s2+s[i].s2;
subtotal.s3=subtotal.s3+s[i].s3;
}
printf("Student Total \n\n");
for(i=0;i<3;i++)
printf("Student %d total is =%d\n",i+1,s[i].total);
printf("Subject Total \n\n");
printf("Subject1 total is =%d\n",subtotal.s1);
printf("Subject 2 total is =%d\n",subtotal.s2);
printf("Subject3 total is =%d\n",subtotal.s3);
getch();
}

```

Output

Result

Thus the C program to calculate subject wise and student wise total was written, compiled, executed and output was verified.

EXP NO : 18 READ TEN VALUES OF ARRAY VARIABLE

DATE:

AIM: To write a C program to read ten values to an array variable and to locate and display value using pointers.

REQUIREMENTS:

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS

• TURBO C

ALGORITHM:

1. Start the program.
2. Declare an integer array and a pointer variable.
3. Initialize the pointer variable with the address of first variable in array.
4. Get ten values from user as input.
5. Using for loop get the values from user.
6. Using pointer variable locate, display the values stored in array with for loop.
7. Stop the program.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[10];
int *p;
int i;
clrscr();
p=&a[0];
printf("Enter 10 integer values on by one \n");
for(i=0;i<10;i++)
scanf("%d",&a[i]);
printf("The output is \n");
for(i=0;i<10;i++)
{
printf("%d\n",*p);p++;
}
getch();
```

```
}
```

OUTPUT:

RESULT:

Thus, the program to read ten values to an array variable and to locate and display value using pointers was written , compiled, executed and output was verified.

TABLE OF INTEGERS

EXP NO :19

DATE :

AIM :

To write a C program to use a table of integers whose size will be specified interactively at runtime.

Requirements

- Computer with PENTIUM IV/DUAL CORE PROCESSORS
- TURBOC

ALGORITHM

- 1.Start the Program
- 2.Get the values for m,n from the user as input.
- 3.Dynamically allocate the size using pointer variable a
- 4.Get the row value & column value.
- 5.Print the values.
- 6.Stop the Program.

PROGRAM

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
int m,n,i,j,*a;
printf("Give the value of row m and column n");
scanf("%d%d",&m,&n);
a=(int*)malloc(m*n*sizeof(int));
for(i=0;i<m;i++)
for(j=0;j<n;j++)
scanf("%d",a+(n*i+j));
printf("the values are");
for(i=0;i<m;i++)
for(j=0;j<n;j++)
printf("%d\n",*(a+(n*i+j)));
}
```

Output

Result

Thus the c program to use a table of integers whose size will be specified interactively at run time was written, compiled, executed and output was verified.

EXP NO:20 STORE CHARACTER STRING USING MALLOC

DATE :

AIM :

To write a c program to store a character string in a block of memory space created by MALLOC and then modify the same to store a larger string.

REQUIREMENTS

- Computer with PENTIUM IV/ DUAL CORE PROCESSORS
- TURBOC

ALGORITHM

- 1.Start the Program
- 2.Declare the pointer variable a.

3. Dynamically allocate the size using pointer variable a using MALLOC function
4. Change the size of previously allocated space using realloc function.
5. Print the values
6. Stop the program.

PROGRAM

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void main()
{
char *a;
a=(char*)malloc(7);
strcpy(a,"welcome");
printf("%s\n",a);
a=(char*)realloc(a,15);
strcpy(a,"welcome you all");
printf("%s",a);
}
```

OUTPUT

Result

Thus the c program to store a character string in a block of memory space created by MALLOC and then modify the same to store a larger string was written, compiled, executed and output was verified.

MALPTC