**ASSIGNMENT – 1**

**1.PROBLEM STATEMENT**

Write a program in C to implement base conversion from any base to any other base (among binary,decimal,octal and hexadecimal).

**2.ALGORITHMS**

Algorithm **Decimal\_To\_Other**

**Input:** A decimal number ‘num’ and the base ‘dbase’ in which it is to be converted.

**Output:** The decimal number ‘num’ converted to base ‘dbase’ and store in an array whose address is returned.

**Remarks:** The input decimal number must be valid.

**Steps:**

1. k=0
2. temp=num
3. **While**(temp≠0) **do**
4. Find the value of temp modulo dbase and store it in ‘rem’
5. Store rem in an array of integers ‘result’ at k
6. Divide temp by 10
7. Increment k by 1
8. **EndWhile**
9. **Return** result

Algorithm **Other\_To\_Decimal**

**Input:** A number ‘num’ and it’s base ‘sbase’.

**Output:** The number ‘num’ converted to a decimal number and returned.

**Remarks:** The input number must be valid.

**Steps:**

1. **Reverse**(num) //reverse the number
2. temp=num
3. sum=0
4. i=0
5. **While**(temp≠0) **do**
6. digit = temp modulo 10 //obtain last digit
7. sum=sum+(digit \* sbasei) //find value
8. temp = temp/10 //eliminate last digit from temp
9. i=i+1
10. **EndWhile**
11. **Return** sum
12. **Stop**

Algorithm **Reverse**

**Input:** The address of the integer array ‘arr’ to be reversed and the number of elements ‘len’.

**Output:** The contents of ‘arr’ reversed.

**Remarks:** The array must be passed as pointer.

**Steps:**

1. **For**(i=1 to len) **do**
2. **arr[i]=arr[i]+arr[j]**
3. **arr[j]=arr[i]-arr[j]**
4. **arr[i]=arr[i]+arr[j]**
5. **EndFor**
6. **Stop**

Algorithm **Other\_To\_Other**

**Input:** A number ‘num’ with its base ‘sbase’ and the base in which it is to be converted ‘dbase’.

**Output:** The number ‘num’ converted to base ‘dbase’ and returned.

**Remarks:** The input number must be valid.

**Steps**

1. res1=**Other\_To\_Decimal**(num,sbase) //find decimal equivalent of num
2. **res2=Decimal\_To\_Others**(res1,dbase) //convert res1 to required base.
3. **Return** res1
4. **Stop**

**3.SOURCE CODE**

#include<stdio.h>

#include<string.h>

#include<math.h>

#include<stdlib.h>

//function to reverse an array

void reverse(int \*arr,int len)

{

    int i,j;

    for(i=0,j=len-1;i<j;i++,j--)

    {

        arr[i]=arr[i]+arr[j];

        arr[j]=arr[i]-arr[j];

        arr[i]=arr[i]-arr[j];

    }

}

//function to convert a number from a given base to decimal

int othertodec(char \*num,int sbase)

{

    int i,x,sum=0;

    strrev(num);

    //Mapping Each Element Of String

    for(i=0;num[i]!='\0';i++)

    {

        if(sbase==16 && num[i]>=65&&num[i]<=70)

            num[i]=num[i]-7; // Joining the number and character sequences together

        num[i]=num[i]-'0';

        sum = sum + num[i]\*pow(sbase,i);

    }

    return sum;

}

//function to convert a decimal number into required base

void dectoother(char \*num,int dbase)

{

    int remarr[20];

    int i,j,k,val=0,tempval,len,rem;

    sscanf(num,"%d",&val); //extract integer from string

    tempval=val;

    i=0;

    while(tempval!=0)

    {

        remarr[i]=tempval%dbase;

        if(remarr[i]<10) //if remainder is less than 10

            remarr[i]+'0'; //add ASCII value of 0

        else if(remarr[i]>9) //if remainder is greater than 9

            remarr[i]+'A'; //add ASCII value of A

        tempval=tempval/dbase;

        i++;

    }

    reverse(remarr,i);

    for(j=0;j<i;j++) //display the result char by char

    {

        if(remarr[j]>9)

            printf("%c",remarr[j]+55);

        else

            printf("%d",remarr[j]);

    }

}

//function to convert a number from one base to another base

void othertoother(char\*num,int sbase,int dbase)

{

    int res1,temp,i=0,j;

    char arr[20];

    res1=othertodec(num,sbase); //find decimal equivalent of num

    temp=res1;

    //converting the decimal equivalent to a string

    while(temp!=0)

    {

        arr[i]=temp%10+'0';

        temp = temp/10;

        i++;

    }

    arr[i]='\0';

    strrev(arr);

    dectoother(arr,dbase);

}

int main(void)

{

    int sbase,dbase,res1;

    char num[20];

printf(“For base conversion of a number: \n);

    printf("Enter Input Base: ");

    scanf("%d",&sbase);

    printf("Enter Output Base: ");

    scanf("%d",&dbase);

    printf("Enter the number: ");

    fflush(stdin); //empty input buffer

    gets(num);

    if(dbase==10)

    {    res1=othertodec(num,sbase);

        printf("DECIMAL:%d",res1);

    }

    else if(sbase==10)

        dectoother(num,dbase);

    else if(sbase!=10 && dbase!=10)

    {

        othertoother(num,sbase,dbase);

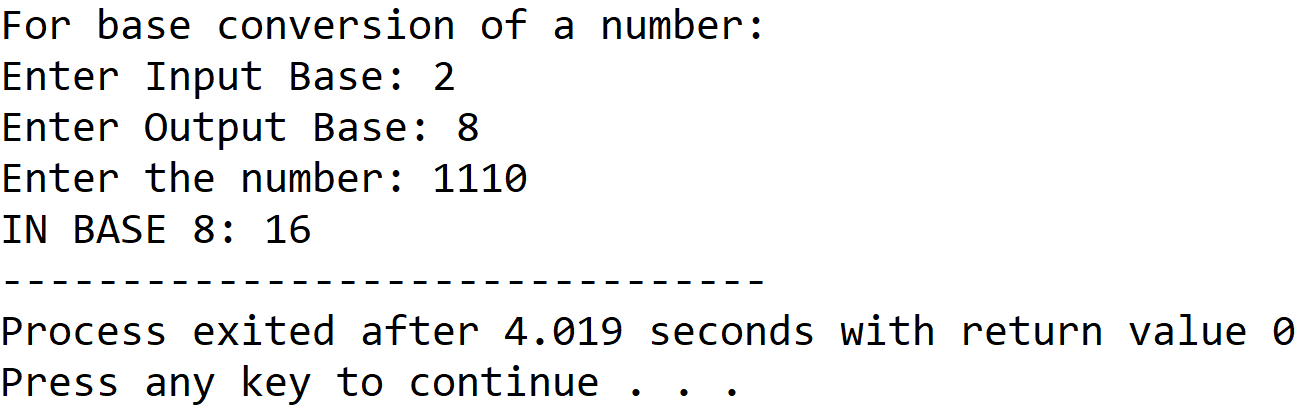
    }

    return 0;

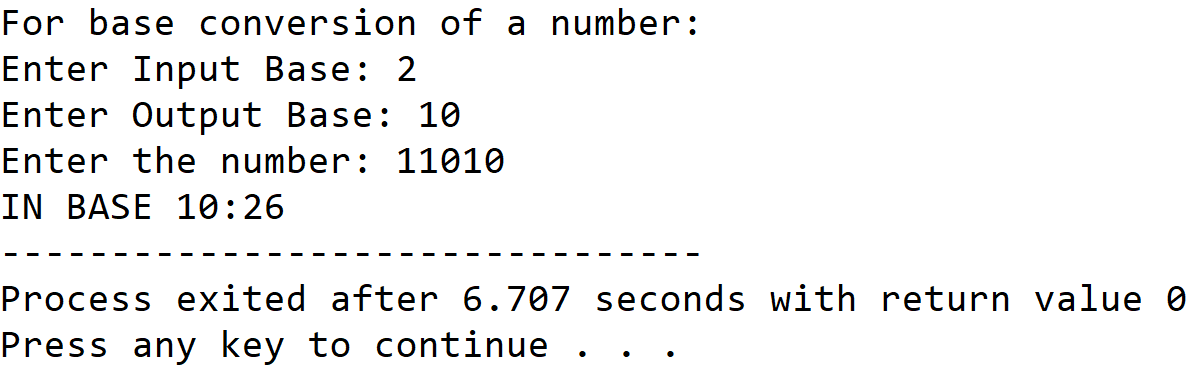
}

**4.OUTPUT**

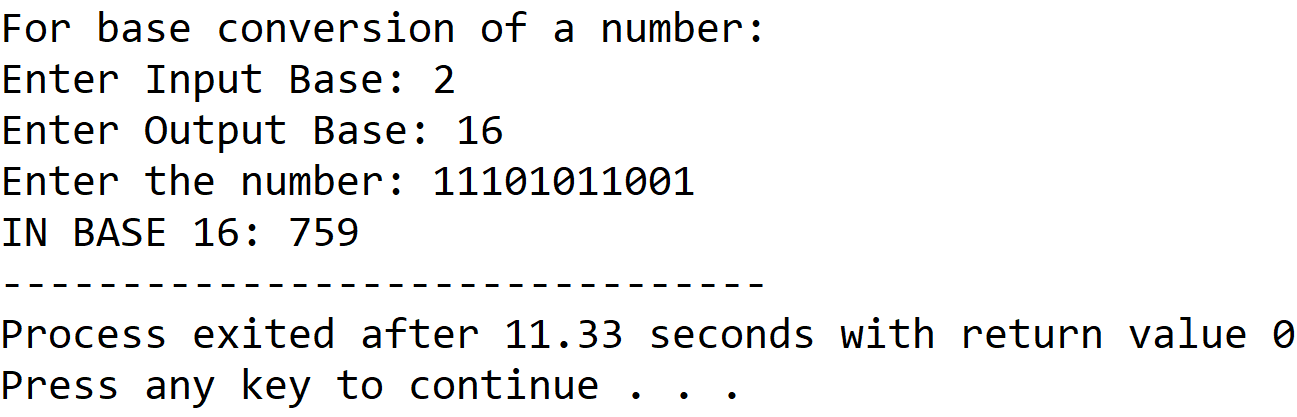
**SET 1:** Binary to octal



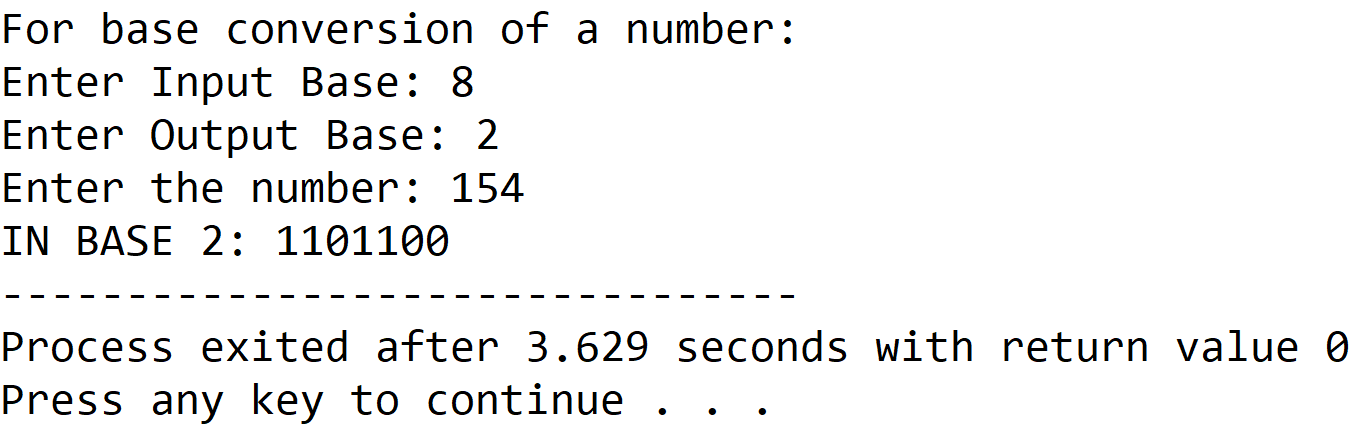
**SET 2:** Binary to decimal



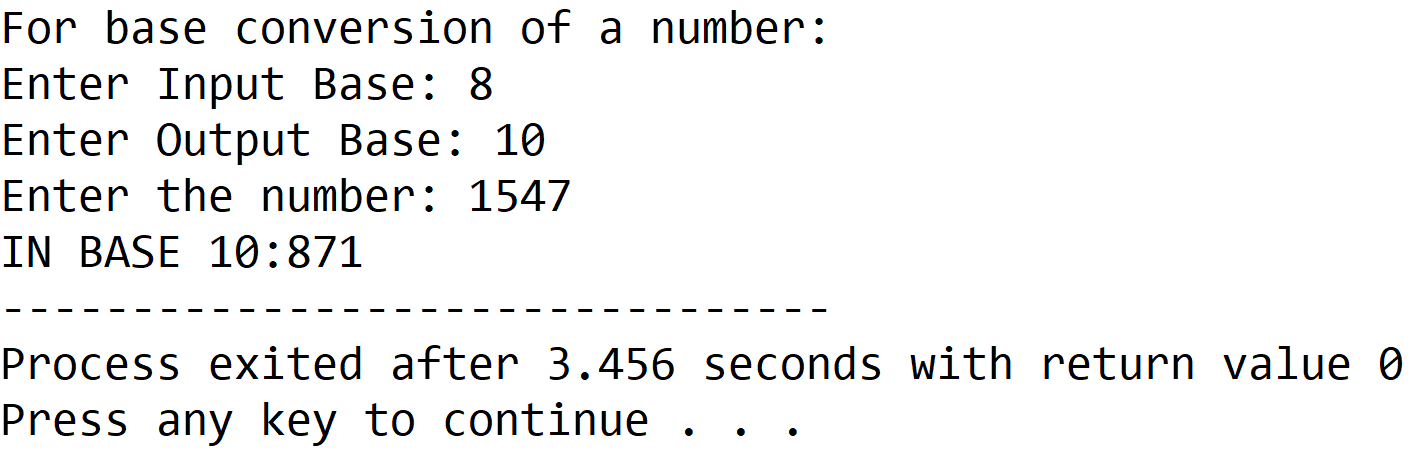
**SET 3:** Binary to hexadecimal



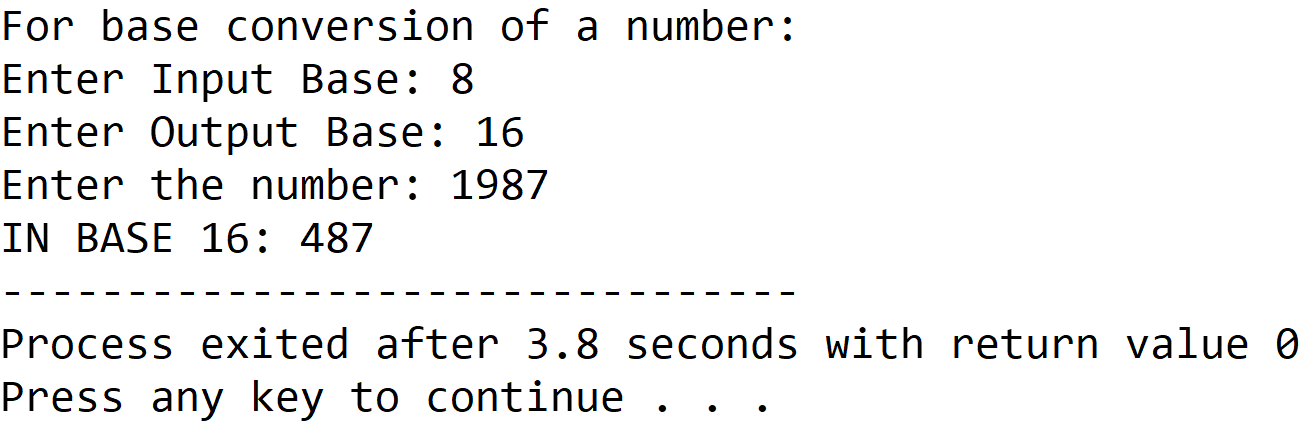
**SET 4:** Octal to binary



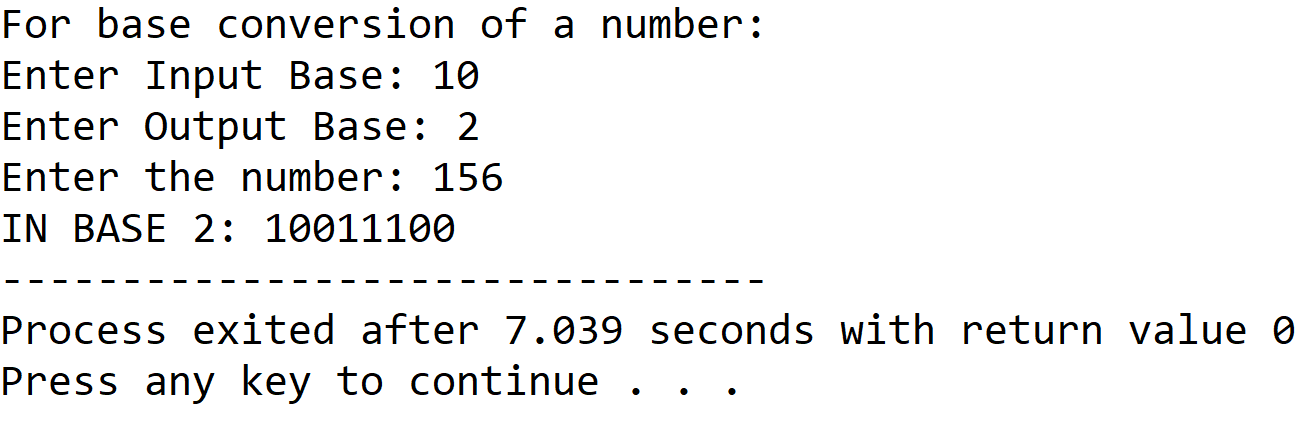
**SET 5:** Octal to decimal



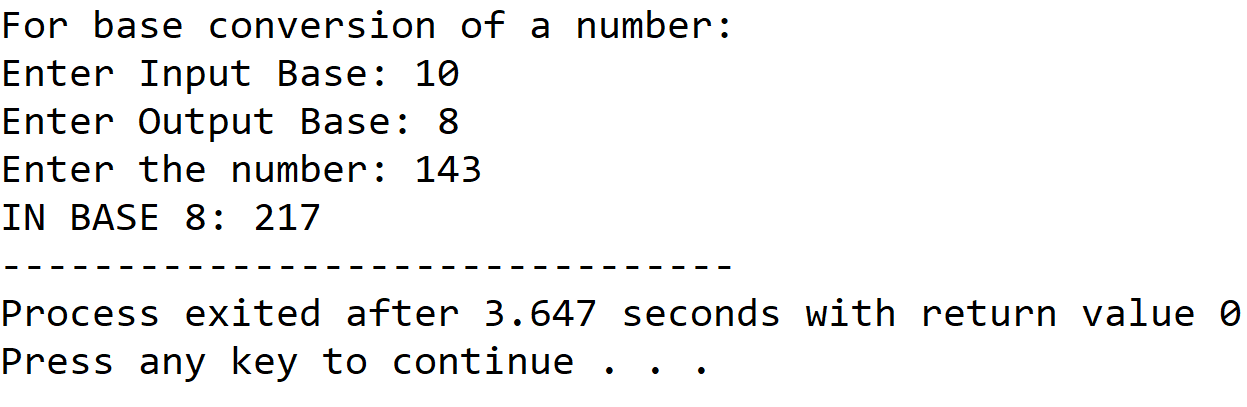
**SET 6:** Octal to hexadecimal



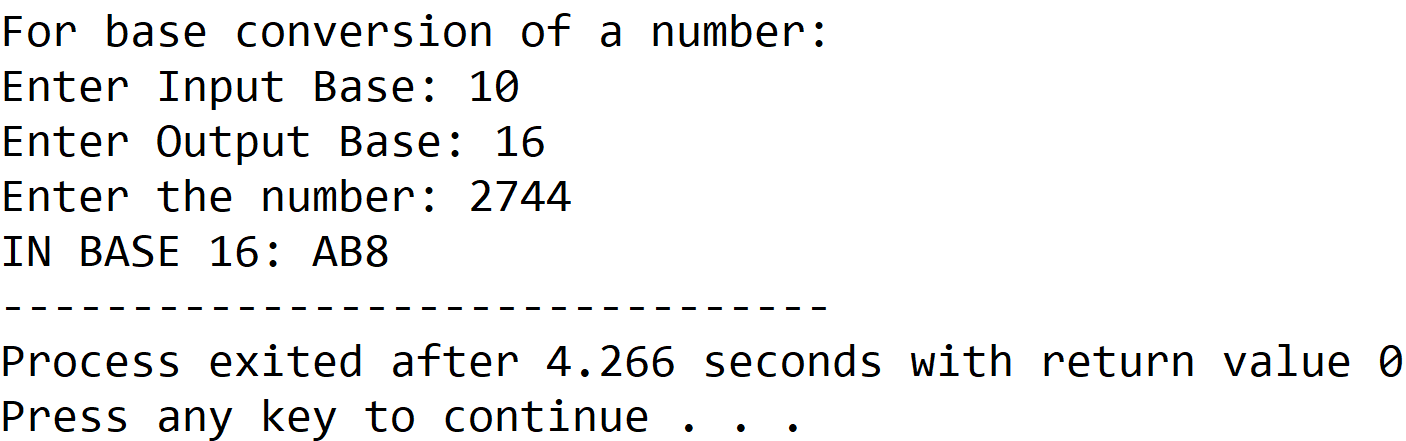
**SET 7:** Decimal to binary



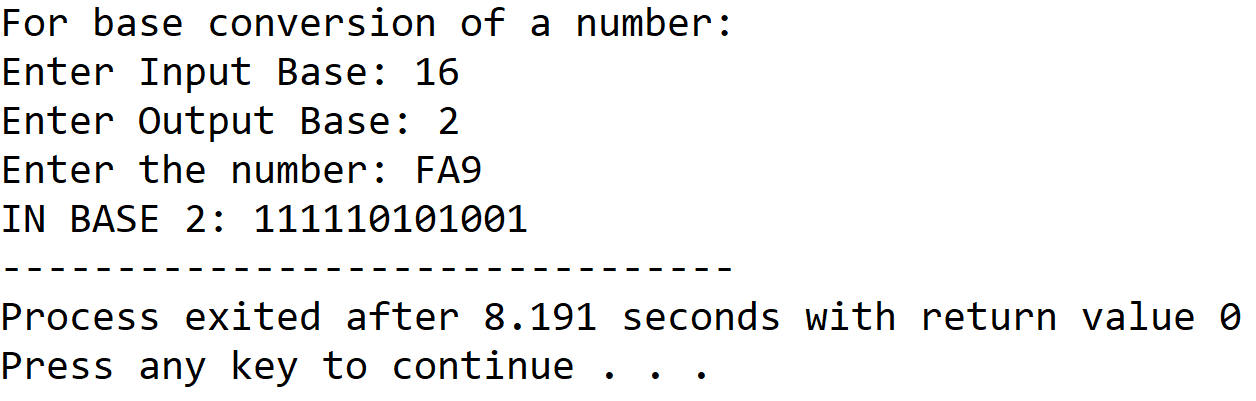
**SET 8:** Decimal to octal



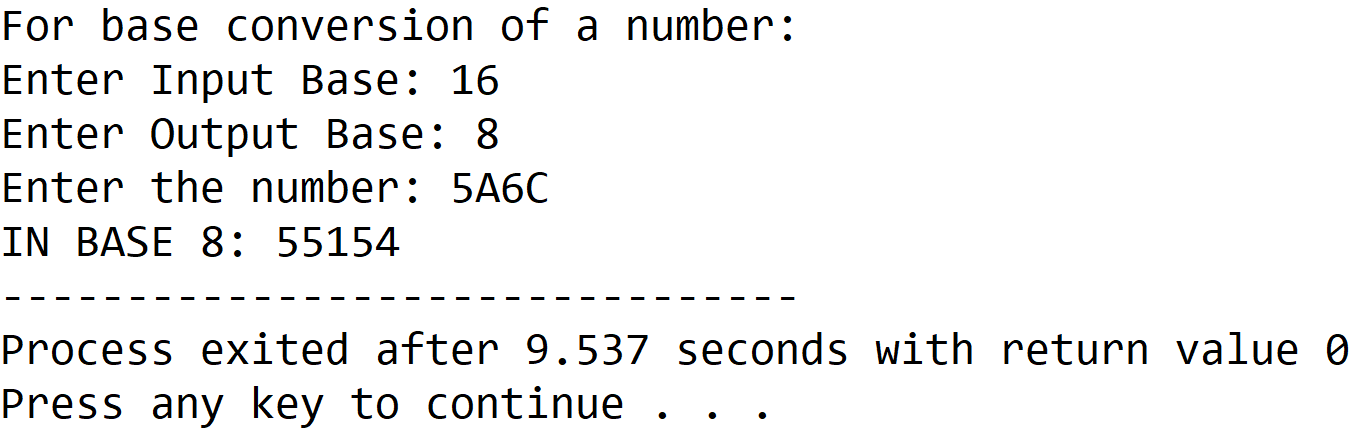
**SET 9:** Decimal to hexadecimal



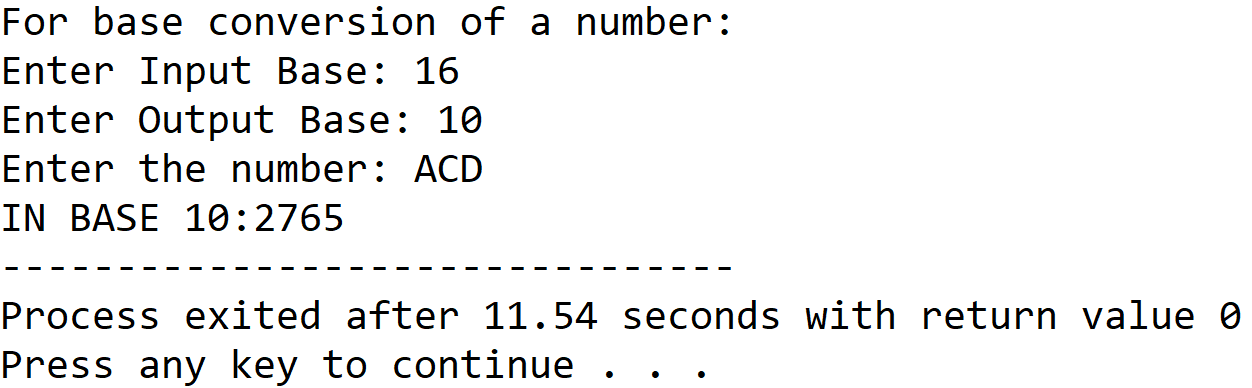
**SET 10:** Hexadecimal to binary



**SET 11:** Hexadecimal to octal



**SET 12:** Hexadecimal to decimal



**5.DISCUSSIONS**

**Variable Description**

* **Sbase:** input base
* **Dbase:** output base
* **Num:** string to hold the input number
* **Remarr:** integer array to hold remainders in dectoother procedure
* **I,j,k:** loop counters

**Limitations**

* There is no procedure to check the user input, so the input must be valid otherwise the program will produce undesired output.

**Uses**

* The program can be used to convert any number among the bases: binary,decimal,octal and hexadecimal. It can find implementation in a calculator program.

**Future Scope**

* A provision for input validation may be added.
* The program can be modified to support more bases.