9. Menu driven code convertor –

binary to decimal and vice versa
binary to hexa and vice versa
binary to octal and vice versa
decimal to hexa and vice versa
decimal to octal and vice versa
octal to hexa and vice versa

codes:

```
ong dec_to_bin(long);
 long dec_to_oct(long);
 long dec_to_hexa(long);
int main()
 {
 int i,j,q;
 printf("Press 1 for DEC to BIN\n");
 printf("Press 2 for DEC to OCT\n");
 printf("Press 3 for DEC to HEXA\n");
scanf("%d",&i);
switch(i)
 {
 case 1:
 long k,j;
 printf("Enter the no.");
 scanf("%d",&j);
 printf("The Equivalent Binary is\n ");
 dec_to_bin(j);
break;
 case 2:
 long y,x;
 printf("Enter the no.");
 scanf("%d",&x);
 printf("The Equivalent octal is\n ");
 dec_to_oct(x);
break;
 case 3:
 long g,h;
 printf("Enter the no.");
 scanf("%d",&h);
 printf("The Equivalent hexadecimal is\n ");
 dec_to_hexa(h);
break;
 default:
 break;
}
system("Pause");
 return 0;
 }
long dec_to_bin(long i)
{
long j[100],k,l=0,m;
 j[0]=k=1;
 while(i!=0)
 {
 j[l]=i%2;
 i=int(i/2);
l++;
 }
for(m=l-1;m>=0;m--)
 {
 printf("%d ",j[m]);
 }
return 0;
 }
 long dec_to_oct(long i)
 {
long j[100],k,l,m;
 j[0]=k=1;
 l=0;
 while(i!=0)
 {
 j[l]=i%8;
i=int(i/8);
l++;
 }
for(m=l;m>=0;m--)
printf("%d",j[m]);
 return 0;
```

```
long dec_to_bin(long);
 long dec_to_oct(long);
 long dec_to_hexa(long);
int main()
 {
 int i,j,q;
 printf("Press 1 for DEC to BIN\n");
 printf("Press 2 for DEC to OCT\n");
 printf("Press 3 for DEC to HEXA\n");
scanf("%d",&i);
switch(i)
 {
 case 1:
 long k,j;
 printf("Enter the no.");
 scanf("%d",&j);
 printf("The Equivalent Binary is\n ");
 dec_to_bin(j);
break;
 case 2:
 long y,x;
 printf("Enter the no.");
 scanf("%d",&x);
 printf("The Equivalent octal is\n ");
 dec_to_oct(x);
break;
 case 3:
```

```
 long g,h;
 printf("Enter the no.");
 scanf("%d",&h);
 printf("The Equivalent hexadecimal is\n ");
 dec_to_hexa(h);
break;
 default:
 break;
}
system("Pause");
 return 0;
 }
long dec_to_bin(long i)
{
long j[100],k,l=0,m;
 j[0]=k=1;
 while(i!=0)
 {
 j[l]=i%2;
 i=int(i/2);
l++;
 }
for(m=l-1;m>=0;m--)
 {
 printf("%d ",j[m]);
 }
return 0;
 }
 long dec_to_oct(long i)
 {
long j[100],k,l,m;
 j[0]=k=1;
 l=0;
 while(i!=0)
 {
 j[l]=i%8;
i=int(i/8);
l++;
 }
for(m=l;m>=0;m--)
printf("%d",j[m]);
 return 0;
 }
long dec_to_hexa(long i)
 {
char j[100],k,l=0,m,n=65;
j[0]=k=1;
 while(i!=0)
 {
 j[l]=i%16;
 i=int(i/16);
l++;
 }
 for(m=l-1;m>=0;m--)
 {
```

```
if(j[m]>=10)
 {
n=n+j[m]-10;
 printf("%c\n",n);
 }
else
 {
 printf("%d\n",j[m]);
 }
 }
 return 0;
 }
```

```
void BinToHex(int bin) { int hex=0, mul=1, count=1, rem; while(bin!=0) { rem = bin%10; hex = hex + (rem*mul); if(count%4==0) { if(hex<10) hexnum[i] = hex+48; else hexnum[i] = hex+55; mul = 1; hex = 0; count = 1; i++; } else { mul = mul*2; count++; } bin = bin/10; } if(count!=1) hexnum[i] = hex+48; if(count==1) i--; }
```

```
int binaryToDecimal(int n)
{

    int num = n;

    int dec_value = 0;



    // Initializing base value to 1, i.e 2^0

    int base = 1;



    int temp = num;

    while (temp) {

        int last_digit = temp % 10;

        temp = temp / 10;



        dec_value += last_digit * base;



        base = base * 2;

    }



    return dec_value;
```

```
}




int octalToDecimal(int n)
{

    int num = n;

    int dec_value = 0;



    // Initializing base value

    // to 1, i.e 8^0

    int base = 1;



    int temp = num;

    while (temp)

    {



        // Extracting last digit

        int last_digit = temp % 10;

        temp = temp / 10;



        // Multiplying last digit with

        // appropriate base value and

        // adding it to dec_value

        dec_value += last_digit * base;



        base = base * 8;

    }
    return dec_value;
}
```