Program: Niyati's code For Paging

Code:

```
#include<stdio.h>
#include<string.h>
#include<math.h>
struct page_table
{
 int pgno;
 int pframe;
 int valid;
};
int power_of_two(int x)
{
   int ans=0;
   while(x!=0)
   {
      x=x/2;
      ans += 1;
   }
   return ans-1;
}
int main()
{
   struct page_table pt[30];
   int psize,page_size,no_of_page,page_bits,offset_bits,main_size,frames;
   int entries_in_pt,pa_bits,la_bits,la_add,digit,mb,kb,i,k=2;
   int binary_num, decimal_num = 0,remainder,num;

   printf("Enter Process Size (KB):");
   scanf("%d",&psize);
   printf("Enter Page Size (bytes):");
```

```
scanf("%d",&page_size);
printf("Enter Main Memory Size(MB):");
scanf("%d",&main_size);
mb=pow(2,20);
kb=pow(2,10);
psize*=kb;
main_size*=mb;
frames=(main_size/page_size);
printf("Number of Frames is: 2^%d \n",power_of_two(frames));

no_of_page=psize/page_size;
printf("Number of pages is %d \n",no_of_page);

pa_bits=power_of_two(main_size);
printf("Bits in Physical address is %d \n",pa_bits);

la_bits=power_of_two(psize);
printf("Bits in Logical address is %d \n",la_bits);

page_bits=power_of_two(no_of_page);
printf("Bits required to find pages %d \n",page_bits);

offset_bits=la_bits-page_bits;
printf("Bits required to find offset %d \n",offset_bits);

//Assume only 10 frames can be allocted to the process
for(i=0;i<no_of_page;i++)
    pt[i].valid=0;

printf("The first 10 entries of Table are: \n ");
printf("Page No  Frame Number Valid \n");
```

```
for(i=0;i<10;i++,k+=2)
{
   pt[i].pgno=i;
   pt[i].pframe=k;
   if(i%2==0)
      pt[i].valid=1;
   printf("%d \t  %d \t\t %d \n",pt[i].pgno,pt[i].pframe,pt[i].valid);
}
printf("Enter logical address: ");
scanf("%d", &num);
binary_num = num >> 5;
i=0;
while(binary_num != 0)
{   remainder = binary_num % 10;
   binary_num /= 10;
   decimal_num += remainder * pow(2, i);
   i++;}
i=0;
do
{
   if(pt[i].pgno==decimal_num)
   {
      if(pt[i].valid==1)
      {
         printf("Page HIT");
      }
      else
         printf("Page FAULT");
      break;
   }
   else
```

```
        i++;
   }while(i!= no_of_page);


   return 0;
}
```

Output:

```
Enter Process Size (KB):8
Enter Page Size (bytes):32
Enter Main Memory Size(MB):2
Number of Frames is: 2^16
Number of pages is 256
Bits in Physical address is 21
Bits in Logical address is 13
Bits required to find pages 8
Bits required to find offset 5
The first 10 entries of Table are:
 Page No  Frame Number Valid
0       2           1
1       4           0
2       6           1
3       8           0
4       10           1
5       12           0
6       14           1
7       16           0
8       18           1
9       20           0
Enter logical address: 0000000000011
Page HIT
```