

TP3

Exerice2:

```
#include <stdio.h>
#include <math.h>
int premier(int n)
{
    int i=2;
    while ((i<n) && (n%i!=0))
    {
        i++;
    }
    if (i<n) return 0;
    else
        return 1;
}

void fact(int n)
{
    int i;
    if(premier(n))
    {
        printf("%d est premier! ",n);
    }
    else
    {
        for ( i = 2; i < sqrt(n)+1; i++)
        {
            if(n%i == 0){
                if (premier(i)) printf("%d\\n",i);
                n = (n/i);
                printf("n= %d\\n",n);
                fact(n);
            }
        }
    }
}

void main()
{
    fact(24);
}
```

Exercice3:

```
#include <stdio.h>

int recherche_dichotomique_recursive(int element,int liste_triee[],int a,int b)
{
    int m,n;
    if (a==b)
    {
        return a;
    }
    if (b==-1)
    {
        b=liste_triee[n-1];
    }
    m = (a+b)/2;
    if (liste_triee[m]==element)
    {
        return m;
    }
    if (liste_triee[m]!=element)
    {
        return recherche_dichotomique_recursive(element, liste_triee, a, m-1);
    }
    else
    {
        return recherche_dichotomique_recursive(element, liste_triee, m+1, b);
    }
}

void main()
{
    int Tab[]={1,2,3,4,5,6};
    int a,b,x,R;
    printf("donner a et b et x \n");
    scanf("%d\t%d\t%d",&a,&b,&x);

    if(recherche_dichotomique_recursive(x,Tab, a, b)) printf("True");
    else printf("False");
    printf("%d",R);
}
```

Exercise5:

```
//tour hanoi
#include<stdio.h>
void deplacer(char a,char b)
{
    printf("\n%c >>> %c",a,b);
}

void tour_hanoi(unsigned n,char a,char b,char c)
{
    if(n==1)
        deplacer(a,c);
    if(n>1)
    {
        tour_hanoi(n-1,a,c,b);
        deplacer(a,c);
        tour_hanoi(n-1,b,a,c);
    }
}

int main()
{
    tour_hanoi(3,'A','B','C');
    return 0;
}
```

Exercie6:

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

int expo_rapide(int x,int n)
{
    if (n==0)
        return 1;
    else
    {
        if(n==1)
            return x;
        if (n%2==0)
        {
            return expo_rapide(x,(n/2))*expo_rapide(x,(n/2));
        }
        else
        {
            return x*expo_rapide(x,((n-1)/2))*expo_rapide(x,((n-1)/2));
        }
    }
}

void main()
{
    int E;
    E=expo_rapide(6,5);
    printf("%d",E);
}
```