

Corrigé test 2 ASDD 2022-2023

Exercice 1 :4pts

1) 0.5pt

Structure de la valeur de la pile

```
#define maxtaille 100
```

```
typedef struct donnee  
{ int val;  
  int indice;  
}donnee;
```

```
typedef struct pile  
{ donnee don[maxtaille];  
  int ind;  
}pile ;
```

2) 1.5pt

```
Void lecturetab(pile *p, int n)  
{donnee x ;int i ;  
  for(i=0 ;i<n;i++)  
  {printf(" tab[%d]=",i); x.indice=i;  
   scanf("%d\n",&x.val);  
   empiler(p,x) ;  
  }  
}
```

3) 2pts

```
Void affiche (pile p)  
{pile pi ; initpile(&pi) ; donnee x ; int i ;  
  printf((" l'indice ?");scanf("%d\n",&i);  
  while(!pilevide(p))  
  { depiler(&p,&x);empiler(&pi, x) ;  
    if( x.indice ==i)  
    {printf(" la valeur de tab[%d]=%d", i,x.val );break;}  
    else if (x.indice <i) {printf(" la valeur de tab[%d]=0", i); break;}  
  }  
  if(pilevide(p)) printf(" la valeur de tab[%d]=0", i);  
  while(!pilevide(pi))  
  {depiler(&pi,&x);empiler(&p, x) ;}  
}
```

Exercice 2 : 6pts1) **0.5**

```

typedef struct Noeud
{
    int val;
    struct Noeud *FG, *FD;
} Noeud;

main() 0.5
{
    Noeud *racine=NULL;
    racine= cree_noeud (6);
    racine->FG = cree_noeud (2);
    racine->FD = cree_noeud (-4);
    racine->FG->FG = cree_noeud (5);
    racine->FG->FD = cree_noeud (1);
    racine->FD->FG = cree_noeud (0);
    racine->FD->FD = cree_noeud (1);
    racine->FD->FD->FG = cree_noeud (3);
    racine->FD->FD->FD = cree_noeud (2);
}

```

2) **1pt**

```

int som(Noeud *a)
{
    if (a == NULL) return 0 ;
    return a->val + som ( a->FG) + som ( a->FD);
}

```

3) **2pts**

```

int som(Noeud *a)
{
    pile P ; initpile(&p) ; som=0 ;
    empiler(&p, a);
    while(!pilevide(p))
    {
        depiler(&p,a);
        while(a!=NULL)
        {
            som=som+ a-> val;
            empiler(&p, a->FD);
            a=a->FG;
        }
    }
    return som;
}

```

4) **2pts**

Première solution

```

int rech_niv ( Noeud *a,int v,int niveau)// dans main à l'appel niveau=0 ou 1
{
    int x,y;
    if (a==NULL)return -1;
    if( a->val== v) return niveau;
    x=rech_niv ( a->FG, v, niveau+1);
    y=rech_niv ( a->FD, v, niveau+1);
    if(x>y && y!=-1)return y;
    if (x!=-1)return x;
    return y;
}

```

4) **2pts**

Deuxième solution avec le parcours par niveau vu en td algorithme itératif

```
int rech_niv ( Noeud *a,int x)
{if (a == NULL) return -1 ;
  else {FileNoeud fil1 ; Fileint fil2 ;
        enfilerN(&fil1,a) ;enfilerI(&fil2,0); int niveau ; // niveau 0 ou 1
        while( !filevideN(fil1) )
        {defilerN(&fil1,a); defilerI(&fil2,&niveau);
          if( a->val== x) return niveau;
          if( a->FG!=NULL) { enfilerN(&fil1,a->FG) ;
                           enfilerI(&fil2,niveau+1);}
          if( a->FD!=NULL) { enfilerN(&fil1,a->FD) ;
                           enfilerI(&fil2,niveau+1);}
        }
  return -1;
}
```