

Corrigé de l'examen (S1)

Exercice 1:

1 / void Supprim(liste *L)

```
{ liste p=*L ; *L=*L->svt ; free(p) ; }
```

2/ void Insere(int num, liste *L)

```
{ liste p=*L ,prd,temp=creern oeud() ; temp->elem=num ;
```

```
if (*L==NULL || num<*L->elem) { temp->svt=*L ; *L=temp ; }
```

```
else { while(p !=NULL && p->elem<num) { prd=p ; p=p->svt ; }
```

```
temp->svt=p ;prd->sv t=temp ;
```

```
}
```

```
}
```

3/ liste Creatliste(int t, liste *L)

```
{ liste tete=NULL,p ;
```

```
while(t>0) { Insere(*L->elem, &tete); Supprim(L) ; t=t-1024 ; }
```

```
Return(tete) ;
```

```
}
```

II/ a) typedef struct ne *liste ;

```
typedef struct ne { int elem ; liste svt ; } noeud ;
```

```
typedef struct nep *listep ;
```

```
typedef struct { char nomp[20] ; long taille ; liste svt ; listep svtp ;} noeudp ;
```

1/ void Ajout(char NP[20], long t, liste *L, listep *tete)

```
{ listep p, temp=creernoeadp() ; temp->taille=t ;strcpy( temp->nomp,NP) ;
```

```
temp->svt=Creatlist(t, L) ; temp->svtp=NULL ;
```

```
if (*tete==NULL) *tete=temp ; /* quand la liste tete est vide */
```

```
else { for (p=*tete; p->svtp!=NULL;p=p->svtp); /* Avancer p sur le dernier element */
```

```

        p->svtp=temp;
    }
}

```

```

2/ void Delete(char NP[20], liste *L, listep *tete)

```

```

{ listep p=*tete, prd; liste R;

```

```

while (p!=NULL && strcmp(p->nomp,NP)!=0) ) { prd=p; p=p->svtp; }

```

```

R=p->svt;

```

```

while (R!=NULL) { Insere(R->elem, L); Supprim(&R); }

```

```

if (p==*tete) *tete= *tete->svtp;

```

```

else prd->svtp=p->svtp;

```

```

free(p);

```

```

}

```

```

typedef struct { char nomp[20]; long taille; } Vecteur;

```

```

main()

```

```

{ Vecteur V[20]; int n, i, tmin; liste L; /* on suppose que la liste L est créée */

```

```

listep tete=NULL, p ;

```

```

for (i=0 ;i<n ; i++) { scanf("%d", &V[i]. taille);

```

```

                gets(V[i].nomp);

```

```

                Ajout(V[i].nomp, V[i].taille, &L, &tete);

```

```

        }

```

```

scanf("%d", &tmin);

```

```

for (p=tete;p!=NULL; p=p->svtp)

```

```

    if (p->taille < tmin) Delete(p->nomp, &L, &tete);

```

```

}

```

Exercice 2:

```
typedef int typelem1;
typedef struct { int elem; int frq ; } typelem2;
pile2 Constpile(pile1 p1)
{ pile2 p2=initpile(); pile1 R=initpile(); int i ; typelem1 x1, y1 ; typelem2 x2 ;
While ( !pilevide(p1) )
{ desempiler(&p1, &x1); i=1;
  while (!pilevide(p1))
    { desempiler(&p1, &y1);
      if (x1==y1 ) i++; else empiler(&R, y1);
    }
  Copiepile(p1, R); /* copie R dans p1*/
  x2.elem=x1 ; x2.frq=i ; empiler(&p2, x2);
}
return(p2);
}
main()
{ pile1 p1; pile2 p2; pile1 S =initpile(), R=initpile(); typelem1 x1; typelem2 x2; int i;
  p2= Constpile(p1);
while( !pilevide(p2))
{ desempiler(&p2, &x2);
  while (!pilevide(S) && x2.elem>sommetpile(S) )
    { desemplier(&S, &x1); empiler(&R, x1); }
  for (i=x2.frq; i!=0; i--) empiler(&S, x2.elem);
  Copiepile(S, R); /* copie R dans S */
}
}
```