

(18-19) Algorithmique et structures de données 2**Corrigé de l'E.M.D**

a) int premier (int n)

```
{int i ;  
  for(i = 2; i < n/2; i++)  
    if ((n % i) == 0)  
      return 0;  
  return 1; }
```

b) int saveT(int T[100])

```
{int i=0, nb ;  
  scanf("%d", &nb) ;  
  while (nb != 999)  
  { if (nb%2==1)  
    {T[i] = nb ;  
     i++ ;  
    }  
    scanf("%d", &nb) ;  
  }  
  return i ;  
}
```

c) void saveFichier()

```
{FILE *f;  
  int taille; int T[100];  
  f = fopen("fichierT", "wb");  
  if (f==NULL)  
    printf("ERREUR");  
  else  
  {taille = saveT(T);  
   fwrite(&taille, sizeof(int), 1, f);  
   fwrite(T, sizeof(int), taille, f);  
   fclose(f);  
  }  
}
```

```
d) typecellule *instete(typecellule *tete, int x)  
{typecellule *p;  
  p= (typecellule *)malloc(sizeof(typecellule));  
  p->val = x;  
  p->suiv = tete;  
  return p;  
}
```

```
typecellule * creation()  
{typecellule *tete = NULL;  
  int n, i, nb;  
  FILE *f;  
  f=fopen("fichierT","rb");  
  if (f==NULL)  
    printf("ERREUR");  
  else  
    {fread(&n, sizeof(int), 1, f);  
      for (i=1; i<=n; i++)  
        {fread(&nb, sizeof(int), 1,f);  
          if (premier(nb))  
            tete = instete(tete, nb);  
        }  
      fclose(f);  
    }  
  return tete;  
}
```

```
e) void affichage(typecellule *tete)  
{typecellule *p;  
  p = tete;  
  while (p != NULL)  
    {printf("%d ", p->val);  
      p = p->suiv;  
    }  
}
```