

(19-20)

Algorithmique et structures de données1

Rattrapage (Durée 1h)

EX1 : Soit une séquence de nombres entiers qui se termine par 999.

On suppose que cette séquence n'est pas vide.

Ecrire un programme C qui affiche la plus grande et la plus petite valeur. (10 pts)

EX2 : Soit un tableau de caractères T[50].

Ecrire un programme qui cherche un élément donné dans ce tableau. Si la recherche est fructueuse, il imprime sa position sinon un message approprié est affiché. (10 pts)

(19-20)

Algorithmique et structures de données 1

Corrigé du rattrapage

EX1 :

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

```
main()
```

```
{ int grand, petit, saisi ;
```

```
printf("Donner un entier ");
```

```
scanf("%d",&saisi); grand = petit = saisi;
```

```
while (saisi != 999)
```

```
{ if (grand < saisi)
```

```
    grand = saisi ;
```

```
    else
```

```
        if (petit > saisi)
```

```
            petit = saisi ;
```

```
printf("Donner un entier ");
```

```
scanf("%d", &saisi);
```

```
}
```

```
printf("le plus grand = %d et le plus petit = %d ", grand, petit);
```

```
}
```

EX2 :

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

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
int main()
```

```
{ char T[10],ch;
```

```
int i, tr, taille;
```

```
printf("donner la chaine\n");
```

```
scanf("%s", T);
```

```
printf("donner la valeur recherchée\n");
```

```
scanf(" %c",&ch);
```

```
tr=0; i=0; taille = strlen(T);
```

```
while (!tr && i < taille)
```

```
    if (T[i]==ch)
```

```
        tr=1;
```

```
    else
```

```
        i++;
```

```
if (tr)
```

```
    printf("%c existe à la position %d", ch, i+1);
```

```
else
```

```
    printf("%c n'existe pas", ch);
```

```
return 0;
```

```
}
```