

<p><b>Exercice 1:</b> <b>(5pts)</b></p> <ol style="list-style-type: none"> <li>14</li> <li>34</li> <li>&amp;A[1]</li> <li>&amp;A[0]</li> <li>&amp;A[1]</li> <li>7</li> <li>&amp;A[2]</li> <li>12</li> <li>169</li> <li>576</li> </ol>	<p><b>Exercice 2: (7.5 pts)</b></p> <pre> #include &lt;stdio.h&gt; #define NMAX 20 main() {int T[NMAX]; int N,i,j,PMIN,AIDE, M; do {printf("Dimension du tableau (max.%d) : \n", NMAX); scanf("%d", &amp;N); }while (N&lt;0    N&gt;NMAX); // saisie des nbrs paires for (i=0; i&lt;N; i++) do{printf("Entrer un nbr paire pour T[%d]: ", i); scanf("%d",&amp;T[i]); }while(T[i]%2); // affichage du tableau for (i=0; i&lt;N; i++) printf("T[%d]= %d \n",i,T[i]); // Tri du tableau par sélection directe du maximum. for (i=0; i&lt;N-1; i++) {PMIN=i; for (j=i+1; j&lt;N; j++) if (T[j]&lt; T[PMIN]) PMIN=j;  if(PMIN!=i) {AIDE= T[i]; T[i]= T[PMIN]; T[PMIN]=AIDE;} } // affichage du tableau trié printf("Le tableau trié dans l'ordre croissant : \n"); for (i=0; i&lt;N; i++) printf("T[%d]= %d \n",i,T[i]); /* on considère le cas le plus simple : on ne tient pas compte de la nature de M et i utile de réallouer de l'espace mémoire pour l'insertion on suppose qu'il y a de l'espace*/ printf("\n Introduire un nombre M à insérer dans le tableau : "); scanf("%d", &amp;M ); // Position d'insertion de M, i=0; while(M&gt;T[i])i++; // Décalage des elements &gt; M for(j=N;j&gt;i;j--)T[j]=T[j-1]; // insertion de M T[j]=M;  printf("Le tableau après insertion : \n"); for (i=0; i&lt;=N; i++) printf("T[%d]= %d \n",i,T[i]);  system("pause");} </pre>	<p><b>Exercice 3: (7.5 pts)</b></p> <pre> #include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; char * recherche(char *s,char c); int compter(char *s,char c); char * Netttoyer(char *s,char c); main() {char chaine[50], car,*ch; int nbr; printf("Entrez une chaine (max.30 caractères) :\n"); gets(chaine);  printf("Entrez un caractère :\n"); scanf("%c",&amp;car); ch = recherche(chaine,car); printf(" la première occurrence de %c est : %s \n",car,ch);  nbr= compter(chaine,car); printf("Le nombre d'occurrences de %c est %d ", car,nbr);  ch= Netttoyer(chaine,car); printf(" la nouvelle chaine après élimination des occurrences de %c est : %s \n",car,ch); system("pause"); } /*****/ char * recherche(char *s,char c) {int i,lg; char *ch=NULL; i=0; while(s[i]!='\0') if(s[i]==c) { ch=&amp;s[i]; break; } else i++; return ch; } /*****/ int compter(char *s,char c) {int nbr=0;  s=recherche(s,c); while(s) {nbr++;s=recherche(++s,c);} return nbr; } /*****/ char * Netttoyer(char *s,char c) {char *P1, *P2; int N; N = strlen(s); for (P1=P2=s; P1&lt;s+N; P1++) {*P2 = *P1; if (*P2 != c) P2++; } /* Nouvelle longueur de s */ N = P2-s; P1=s; *(P1+N)='\0'; return P1;} </pre>
---	---	---

