

CORRECTION SESSION NORMALE

PROGRAMMATION C

INF 131 (2021~2022)

Proposez par : GROUPE GENIUS REPETITION

Par : Joël_yk

Exercice 01 : 05 pts

VOIR CORRECTION INTEGRALE DU CC 2021.

Exercice 02 : 03 pts

1) Fonction : factoriel 1pts

```
double factoriel(int a)
{
    if(a == 0 || a == 1)
        return 1;
    else
        return a*factoriel(a-1);
}
```

2) Fonction : puissance 1pts

```
int puissance(int a , int b ) {
// a^b ce lit a puissance b
    if( b==0 )
        return 1;
    else return a * puissance(a,b-1);
}
```

3) Programme SINUS(X) 1pts

```
#include <stdio.h>
# Define epsi 0.001 ;
```

```

int main() {
    float x;
    printf("Veuillez saisir la valeur de x\n");
    scanf("%d",&x);
    double sinus = 0 ;
    do{
        sinus += (puissance(-1,i)*puissance(x,2*i+1))/ factoriel(2*i+1);
        i++ ;
    } while ( sinus > x*epsi ) ;
    printf("Le sinus de %d est : %d\n",x,sinus);
    return 0;
}

```

Exercice 03 : 06 pts

1) Les structures de données : 3pts

```

#define N 10000
#define M 500
typedef struct Approvi {
    int qteProd;
    int dateAppro; // vous pouvez déclarer une Structure Date tel que Date dateAppro
    float prixAchat;
} Approvi;

typedef struct produit {
    char code[5];
    char* designation; // ou char designation[20]
    char* categorie; // ou char categorie[20]
    float prixVente;
    int qteAlerte;
    int nbreAppro;
    Approvi listeAppro [M] ;
} produit;

```

2) Ecriture de la fonction Existence :

```

int Existence(char*code, produit prod [ ] ) {

```

```

for(int i = 0; i<N; i++) {
    if(strcmp(prod[i].code,code))
        return 0; } else {return 1; } }

```

3) Ecriture de la fonction ajoutez Produit :

```

void AjoutProduit(produit P [ ] )
{
    printf("Veuillez saisir la position du produit a insérer dans le tableau\n");
    int k;
    scanf("%d",&k);
    printf("Veuillez saisir les informations du produit a insérer\n");
    char* code;
    printf("Veuillez saisir le code\n");
    scanf("%s",code);
    if(Existence(code,P))
        return ;
    else{
        strcpy(P[k].code,code);
        printf("Veuillez saisir la designation\n");
        scanf("%s",P[k].designation);
        printf("Veuillez saisir la categorie\n");
        scanf("%s",P[k].categorie);
        printf("Veuillez saisir le prix de vente\n");
        scanf("%f",&P[k].prixVente);
        printf("Veuillez saisir la quantite d'alerte pour le réapprovisionnement\n");
        scanf("%d",&P[k].qteAlerte);
        printf("Veuillez saisir la quantite de produit\n");
        scanf("%d",&P[k].listeAppro[P[k].nbreAppro].qteProd);
        printf("Veuillez saisir la date de l'approvisionnement\n");
        scanf("%d",&P[k].listeAppro[P[k].nbreAppro].dateAppro);
        printf("Veuillez saisir le prix d'achat\n");
        scanf("%f",&P[k].listeAppro[P[k].nbreAppro].prixAchat);
        P[k].nbreAppro++;
    } }

```

Exercice 04 : 06 pts

1) Programme recherche Sentinelle (sur un tableau non trié T) : 2pts

```

#include <stdio.h>
#define SIZE 30
#define SIZE_S 31
int main() {
    int nbr, i, k, T[SIZE_S];

    printf(" Entrez les éléments du tableau : ");
    for (i = 0; i < SIZE ; i++) {
        scanf("%d", &T[i]);
    }
    printf(" Entrez l'élément à rechercher: ");
    scanf("%d", &k);

    //Placer la sentinelle
    T[SIZE_S] = k ;

    //La recherche commence à partir de zéro
    i = 0;
    while (i < SIZE && T[SIZE_S-1] != T[i]) {
        i++;
    }
    if (i == SIZE) {
        printf("L'élément se trouve dans la position = %d", i + 1);
    } else {
        printf("Elément non trouvé");
    }

    return 0;
//www.pandacodeur.com
}

```

2) Programme Tri Sélection (sur un tableau non trié T) : 2pts

```

#include <stdio.h>
#define SIZE 30
int main()
{ // Le tri des 03 frères
    int i, j, tmp, k, T[SIZE];

    printf(" Entrez les éléments du tableau : ");
    for (i = 0; i < SIZE ; i++) {

```

```

    scanf("%d", &T[i]);
}
for (i=0; i < (SIZE-1); i++) { // (size-1) car un tableau d'un seul élément est déjà trié !
    j = i;
    for (k=i + 1; k < SIZE; k++) {
        if (T[j] > T[k])
            j = k;
    }
    if (j != i) { // permet d'éviter les permutations inutiles
        tmp = T[i];
        T[i] = T[j];
        T[j] = tmp;
    }
}

printf("\n**** Le tableau triée par ordre croissant ****\n");

for (i=0; i < SIZE; i++)
    printf("%4d", T[i]);

return 0;
//www.pandacodeur.com
}

```

3) Programme recherche Dichotomique (sur un tableau trié T) : 2pts

```

#include <stdio.h>
#define SIZE 30
int main() {
    /* Déclarations */
    int T[SIZE]; /* tableau donné */
    int k; /* valeur à rechercher */
    int I; /* indice courant */
    int POS; /* position courantes */
    int INF, MIL, SUP; /* limites du champ de recherche */
    //réalisez par le groupe genius

    /* Saisie des données */
    printf("Entrer Elément à rechercher : ");
    scanf("%d", &k );

    /* Affichage du tableau suppose trié par ordre croissant */
}

```

```

printf(" Le Tableau donné : \n");
for (I=0; I< SIZE; I++)
    printf("%d ", T[I]);
printf("\n");

/* Initialisation des limites du domaine de recherche */
INF=0;
SUP= SIZE- 1;
/* Recherche de la position de la valeur */
POS=-1;
while ((INF<=SUP) && (POS== - 1)) {
    MIL=(SUP+INF)/2;
    if ( k < T[MIL])
        SUP=MIL-1;
    else if (k > A[MIL])
        INF=MIL+1;
    else
        POS=MIL;
}

/* Affichage du résultat */
if (POS== - 1)
    printf("La valeur recherchée ne se trouve pas "
        "dans le tableau.\n");
else
    printf("La valeur %d se trouve à la position %d. \n",
        k , POS);

return 0;
//www.pandacodeur.com
}

```

"La persévérance, c'est ce qui rend l'impossible possible, le possible probable et le probable réalisé."

Bonne chance pour le rattrapage les amies.

Contact WhatsApp : +237 658³⁹59⁷⁸ | Réaliser Par Joël_Yk .