UE[HT-P1-INFRES-002-M] :

*Nom: ........................................*

*Prénom: ....................................*

*Groupe: .....................................*

Programmation - Bases et Algorithmique – G1

|  |  |
| --- | --- |
| **Question théorique :** | **Sous total : /30** |

Questionnaire à choix multiples, entourez **la** bonne réponse. Une bonne réponse vaut 2 points, une mauvaise réponse vaut -1, ne rien répondre ne vaut aucun point. Pour le code on considère qu’il est dans une fonction et les bibliothèques sont incluses. Vous avez 50 minutes pour la théorie.

|  |  |
| --- | --- |
| **Questions** | **Réponses** |
| 1. Laquelle de ces propositions ne fait pas parti de l’architecture de Von Neumann ? | 1. L’unité arithmétique et logique 2. L’unité de contrôle 3. L’unité vectorielle 4. La mémoire |
| 1. Laquelle de ces propositions ne fait pas parti des structures de données en algorithmique ? | 1. Constantes 2. Variables 3. Séquences 4. Tableaux |
| 1. Lequel de ces noms de variables n’est pas permis en C? | 1. annee2015 2. annee\_2015 3. annee 2015 4. Annee2015 |
| 1. Lequel de ces types prend le moins de place en mémoire ? | 1. short 2. int 3. short int 4. unsigned char |
| 1. Qu'affichera le code suivant ?   char a='a' ;  a=a+3 ;  //a='z' ;  printf("%c a %c",a,'a') ; | 1. c a a 2. 3 a a 3. a a a 4. d a a 5. c a c |
| 1. Que se passe-t-il lorsqu’on déclare une variable comme ceci :   unsigned int x ; | 1. La variable x prend 2 fois moins de place en mémoire 2. La variable x prend un octet de moins en mémoire 3. La variable x peut stocker des valeurs positives 2 fois plus grandes 4. La variable x prend 1 bit de moins en mémoire. |
| 1. Qu’affichera le code suivant ?   int main() {  int x=1,y=2;  y=x\*y;  x=x\*y;  x=x\*y;  printf("%d %d",x,y);  return 0;  } | 1. 4 4 2. 4 2 3. 4 8 4. 4 6 |
| 1. Quels opérateurs agissent sur des valeurs (entières, réelles ou caractères) et donnent des résultats booléens ? | 1. Les opérateurs d'affectation élargie comme += \*= -= /= %= 2. Les opérateurs logiques comme || && 3. Les opérateurs de comparaison comme <><= >= == 4. De tels opérateurs n'existent pas 5. Les opérateurs arithmétiques comme + - \* / % |
| 1. Quelle est la syntaxe pour inclure la bibliothèque string.h ? | 1. #include(string); 2. #include<string.h> 3. #include(string.h); 4. include(string); |
| 1. Qu'affichera le code suivant ?   int x=10;  if (x<10) printf("1");  else printf("2");  printf("3"); | 1. 13 2. 1 3. 123 4. 23 5. 3 |
| 1. De quel type ne peut pas être une variable testée dans une structure conditionnelle SWITCH ? | 1. float 2. int 3. aucun type de variable n’est proscrit 4. long |
| 1. A quoi sert l’instruction CONTINUE ? | 1. à passer à l’itération suivante 2. à mettre fin à un bloc d'instructions 3. à continuer le programme 4. à détruire une variable afin d'en libérer l'espace mémoire |
| 1. Qu'affichera le code suivant ?   int i,x=1;  for (i=1 ; i<5 ; i++) x=(i+1)\*2;  printf("%i",x); | 1. 5 2. 11 3. 1 4. 10 |
| 1. Soit un programme contenant les lignes suivantes :   int i = 0;  int j = 0;  for (i = 0; i < 2; i = i + 1)  {  for (j = 0; j < 3; j = j + 1)  {  printf("%d ", j);  }}  qu'est ce qui sera affiché ? | 1. 1 2 1 2 3 2. 0 1 2 0 1 2 3. 0 0 0 1 1 1 4. 0 1 0 1 0 1 0 1 |
| 1. Soit year[], un tableau de 25 entiers contenant l'année de naissance de 25 étudiants. Quel code permet d'afficher toutes les années de naissance inférieures ou égales à 1990 ? | 1. for (i=0 ; i>=25 ; i++) {   if (year[i]<=1990) printf("%d",year[i]);  }   1. for (i=25 ; i>0 ; i--) {   if (year[i]<=1990) printf("%d",year[i]);  }   1. for (i=25 ; i>0 ; i++) {   if (year[i]<=1990) printf("%d",year[i]);  }   1. for (i=24 ; i>=0 ; i--) {   if (year[i]<=1990) printf("%d",year[i]);  } |
| 1. Soit quantites[], un tableau de 365 entiers contenant des quantités d'articles vendus. Quel code permet de déterminer la quantité totale d’articles vendus ? | 1. int total=0;   for (i=0 ; i<365 ; i++) {  if (quantites[i]>total) total=quantites[i];  }   1. int total=0;   for (i=0 ; i<365 ; i++) {  total=quantites[i];  }   1. int total=0;   for (i=0 ; i<365 ; i++) {  total+quantites[i];  }   1. int total=0;   for (i=0 ; i<365 ; i++) {  total=total+quantites[i];  } |
| 1. Qu'affichera le code suivant ?   char mot[32]="castor";  int i,x;  for (i=0 ; mot[i]!='\0' ; i++) {  printf("%c",mot[i]);  mot[i]++;  } | 1. castor 2. bdtups 3. astor 4. castorcastorcastorcastorcastorcastor |
| 1. Pour affecter la lettre h à une variable, quelle est la bonne syntaxe ? | 1. lettre=(h); 2. lettre=h; 3. lettre='h'; 4. lettre="h"; |
| 1. Quel convertisseur permet d'afficher une structure ? | 1. Le convertisseur dépend du type des paramètres de la structure, aucun convertisseur ne convient à l'affichage d'une structure dans son intégralité 2. %c 3. %s 4. %i |
| 1. Qu'affichera le code suivant ?   int x=5;  int \*p;  p=&x;  printf("%d",&p); | 1. p 2. 5 3. x 4. L'adresse de p |

*Nom: ........................................*

*Prénom: ....................................*

*Groupe: .....................................*

|  |  |
| --- | --- |
| **Question sur ordinateur :** | **Sous total: /70** |

Le programme à réaliser doit permettre l'encodage d'une série de nombres entiers. Les nombres doivent être mémorisés dans un tableau à deux dimensions qui doit obligatoirement être déclaré dans le corps du programme principal (main) et dont les dimensions maximales seront 25 par 25.

L'utilisateur doit pouvoir ensuite décider quelle portion de ce tableau il souhaite remplir (au minimum 5\*5 lignes/colonnes, maximum 25\*25 colonnes. Veillez à prévoir ici un contrôle de saisie sur le nombre fourni par l'utilisateur.

Le programme doit ensuite remplir de manière automatique cette portion de tableau en y insérant des nombres choisis aléatoirement dans l'intervalle 5-50.

Une fois le remplissage achevé, le programme nettoie l'écran puis y affiche les nombres mémorisés sous forme de grille (conserver une seule décimale pour l'affichage).

Le programme doit ensuite permuter les contenus des cellules dans le tableau en prenant comme axe une diagonale descendante : le contenu de la cellule tout en bas à gauche devenant celui de la cellule tout en haut à droite et inversement. Il en va de même pour les autres cellules (voir illustration en salle)

Le contenu du tableau doit ensuite être une nouvelle fois montrée à l'écran.

Finalement le programme doit demander à l'utilisateur de saisir un nombre entier (compris entre 5 et 50). Il doit ensuite déterminer combien de nombres dans le tableau sont strictement supérieurs à ce nombre.

Votre programme doit être décomposé en plusieurs de sous programmes (procédures et/ou fonctions). Le programme principal doit impérativement se présenter de la manière suivante :

**main()**

**{**

**// affichage de votre nom, prénom, classe et groupe**

**// déclaration du tableau 2dimensions destiné à contenir les nombres**

**// déclaration de la variable contenant le nombre ligne et colonnes à remplir (par exemple : "nbr\_ligcol")**

**// appel de la fonction de saisie du nombre de ligne et col. à remplir avec mise à jour de la variable " nbr\_ligcol"**

**// appel de la procédure de remplissage automatique avec valeurs aléatoires**

**// appel de la procédure d’affichage du contenu du tableau**

**// appel de la procédure de permutation du contenu suivant diagonale avec mise à jour du tableau**

**// appel de la procédure d’affichage du contenu du tableau**

**// appel de la fonction de recherche du nombre de valeurs supérieures à une valeur s donnée par l'utilisateur (l’affichage du résultat obtenu doit se faire à l’intérieur du programme principal)**

**}**

***Lorsque vous aurez terminé, vous devrez me remettre un dossier portant votre nom et prénom dans lequel se trouveront le code source commenté de votre programme ainsi que le fichier exécutable créé.***

***Durée maximale de l’épreuve : 2H30***

***Bon travail.***