INF147 – Progammation procédurale

Service des enseignements généraux.

Francis Bourdeau

Laboratoire 02

### Objectif

Ce laboratoire se compose de 15 exercices. Ceux-ci doivent tous être complétés d’ici la prochaine séance de laboratoire. Les notions validées dans ce laboratoire sont :

* L'inclusion de libraire.
* L'écriture de programmes principaux.
* La déclaration et l'emploi de variables; de constantes; et l'emploi des types appropriés.
* Les opérations de calculs et l'emploi de fonction mathématique usuelle.
* L'utilisation d'instructions conditionnelles.

### Classification des exercices

Les exercices de ce laboratoire sont regroupés en trois catégories :

|  |  |
| --- | --- |
| Exercices qui développent des automatismes | |
|  | Ces exercices sont simples, et volontairement répétitifs. Ces exercices servent à répéter des syntaxes simples de manière qu’elles deviennent des automatismes. Si vous arrivez à effectuer ces exercices rapidement, sans regarder vos notes, vous maîtrisez ce type d’exercices, sinon, continuez à répéter. |
| Exercices d’exploration | |
|  | Ces exercices ne sont pas difficiles en soi, mais vous amenez à tester votre compréhension des notes de cours. Après une révision des notes de cours, ces exercices devraient se faire facilement. |
| Exercices de dépassement | |
|  | Ces exercices sont difficiles et ont pour but de vous amener comprendre les notions enseignées en profondeur et de développer les concepts plus avancés associés avec les structures montrées en classe et en exemple.  Même avec les notes de cours, ces exercices peuvent être difficiles. Si vous restez bloqués, réalisez un schéma qui montre l'évolution des valeurs intermédiaires du problème afin d'arriver progressivement vers la solution. |

### Exercices sur les opérateurs logiques

### A picture containing text, clipart Description automatically generated Exercice 01

Écrivez un programme qui saisit le poids et l'âge d'un participant. Pour que ce participant puisse sauter en parachute, il doit être âgé d'au moins 15 ans et peser au minimum 40kg. Afficher si le participant a le droit de sauter en parachute ou non.

Background pattern

Description automatically generated with low confidence

### A picture containing text, clipart Description automatically generated Exercice 02

Écrivez un programme qui saisit deux nombres. Le programme affiche si les deux nombres sont pairs, si les deux nombres sont impairs, ou si l'un est pair et l'autre est impair. Pensez à l'opérateur modulo pour déterminer si un nombre est pair.

Background pattern

Description automatically generated with low confidence

### Icon Description automatically generated Exercice 03

Écrivez un programme qui saisit trois coordonnées qui représentent les trois sommets d'un triangle. Le programme détermine si le triangle est équilatéral (trois côtés égaux), isocèle (deux côtés égaux) ou scalène (aucun côté égal). La longueur d'un côté du triangle se calcule par la formule :

Text

Description automatically generated

### Exercices sur les boucles for

### A picture containing text, clipart Description automatically generated Exercice 04

Écrivez un programme qui demande à l’usager d'entrer une valeur limite. Le programme affiche tous les nombres pairs compris entre 0 et cette limite.

Background pattern

Description automatically generated with low confidence

### Exercice 05



Écrivez un programme qui demande à l’usager d'entrer une valeur. Le programme affiche tous les diviseurs de la valeur entrée. Pour trouver les diviseurs de la valeur, il faut tester chacun des diviseurs potentiels entre 2 et , et déterminer si l’un d’entre eux est un diviseur de n en utilisant le modulo.

Background pattern

Description automatically generated with low confidence

### Icon Description automatically generated Exercice 06

Écrivez un programme qui demande à l’usager d'entrer une valeur. Le programme détermine si le nombre est premier ou non. Pour savoir si un nombre est premier, il faut compter le nombre de diviseur que possède cette valeur. Si le nombre possède exactement 2 diviseurs, il est premier.

Background pattern

Description automatically generated with low confidence

### Exercice 07



Écrivez un programme qui demande à l’usager d'entrer une valeur limite. Le programme détermine tous les nombres premiers entre 1 et cette valeur.

A picture containing text

Description automatically generated

### Exercices sur les boucles while

### A picture containing text, clipart Description automatically generated Exercice 08

Écrivez un programme qui demande à l’usager d'entrer une valeur entre 1 et 100. Tant que la valeur lue n'est pas dans l'intervalle, le programme affiche un message d'erreur et recommence.

A picture containing background pattern

Description automatically generated

### Icon Description automatically generated Exercice 09

Écrivez un programme qui demande à l’usager d'entrer une valeur. Le programme détermine le nombre de chiffres présent dans cette valeur.

Stratégie : Comptez le nombre de fois que l'on peut extraire les unités d'un nombre (à l'aide d'un modulo) et se débarrasser de ces unités (à l'aide d'une division entière).



### Exercice 10



Écrivez un programme qui approxime la valeur de arctan(x) en utilisant la formule suivante :

Théoriquement, il faudrait faire accumuler une infinité de terme pour arriver à une valeur précise. Afin de ne pas boucler à l'infinie, utilisez 100 termes dans l'approximation. 

### A picture containing plant Description automatically generated Exercice 11

Écrivez un programme qui approxime π en utilisant le produit de Wallis.

Théoriquement, il faudrait faire accumuler une infinité de terme pour arriver à une valeur précise. Afin de ne pas boucler à l'infinie, continuez jusqu'à ce que le facteur de modification soit inférieur à 0,00001.



### Introduction au nombre aléatoire

Pour réaliser les trois exercices qui suivent, vous aurez besoin de la fonction nb\_aleatoire. Le code qui suit montre comment utiliser cette fonction.

|  |
| --- |
| #include <time.h> /\* Pour avoir accès à time. \*/  #include <stdlib.h> /\* Pour avoir accès à srand. \*/  #include <stdio.h> /\* Pour avoir accès à printf. \*/  /\*  NB\_ALEATOIRE  Cette fonction retourne une valeur aléatoire dans l'intervalle entier  [min, max]. Chaque valeur possède la même probabilité d'être générée.    PARAMÈTRES :  - min : La plus petite valeur pouvant être générée (type : int).  - max : La plus grande valeur pouvant être générée (type : int).    VALEUR DE RETOUR : Une valeur se trouvant dans l'intervalle  [min, max] (type : int).    ANTECEDENTS :  - On suppose que la fonction srand a préalablement été  appelé.  \*/  int nb\_aleatoire(int min, int max)  {  return min + (int)(rand() / (RAND\_MAX + 0.001) \* (max - min + 1));  }  int main(void)  {  int nombre\_aleatoire; /\* Reçoit un nombre aléatoire. \*/  /\* On initialise le générateur de nombres aléatoires. \*/  srand((unsigned int) time(NULL));  rand();  /\* On génère un nombre dans l’intervalle [1, 10]. \*/  nombre\_aleatoire = nb\_aleatoire(1, 10);  /\* On affiche le nombre généré. \*/  printf("Nombre : %i\n", nombre\_aleatoire);  return EXIT\_SUCCESS;  } |

### A picture containing text, clipart Description automatically generated Exercice 12

Écrivez un programme qui lance cinq dés à 6 face et affiche les cinq valeurs obtenues.

Background pattern

Description automatically generated with low confidence

### Icon Description automatically generated Exercice 13

Écrivez un programme qui lance un dé à 6 faces un million de fois et affiche la moyenne des lancers.



### Icon Description automatically generated Exercice 14

Écrivez un programme qui estime la probabilité d’obtenir 7 lorsqu’on lance deux dés à 6 faces et que l’on fait la somme du résultat obtenu. Tentez l'expérience un million de fois, comptez le nombre de fois que les 7 ont été observés et divisés ce nombre d'observations par le nombre de tentatives.

