**Solution**

Exercise 1

#include <stdio.h>

Part 1:

void main()

Test =1 c.à.d. tout nombre par défaut est premier

{

int n, i, test = 1;

printf("donner un nombre pour tester si premier ou pas : ");

scanf("%d", &n);

Test =0 c.à.d. si ce nombre est divise au moins une seule fois par un autre nombre entre 2 et n/2 >> ce nombre devient non premier

for (i = 2; i < n / 2; i++)

if (n%i == 0)

test = 0;

if (test == 0)

printf("ton nombre \"%d\" est non premier", n);

Affichage selon le cas...

else

printf("ton nombre \"%d\" est premier", n);

}

Exemple d’exécution :

**donner un nombre pour tester si premier ou pas : 27**

**ton nombre "27" est non premier**

**donner un nombre pour tester si premier ou pas : 23**

**ton nombre "23" est premier**

Part 2:

void main()

Test =1 c.à.d. tout nombre par défaut est spécial

{

int n, i, j, test = 1;

printf("donner un nombre pour tester si special ou pas : ");

scanf("%d", &n);

Calcul de diviseur

for (i = 2; i < n; i++)

if (n%i == 0)

Test sur chaque diviseur si elle est premier ou pas, si test=0 alors il existe au moins un diviseur qui n’est pas premier c.à.d. on ici un cas d’un nombre non spécial

for (j = 2; j < i / 2; j++)

if (i%j == 0)

test = 0;

if (test == 0)

printf("ton nombre \"%d\" est non special", n);

Affichage selon le cas...

else

printf("ton nombre \"%d\" est special", n);

}

Exemple d’exécution :

**donner un nombre pour tester si special ou pas : 14**

**ton nombre "14" est special**

**donner un nombre pour tester si special ou pas : 20**

**ton nombre "20" est non special**

Part 3:

void main()

{

int n, i, j, test;

printf("les premier 100 nombre speciaux sont :\n");

* Même que partie 2 mais au lieu de scan de n on met n dans un boucle de 0 à 100
* Tout initialisation d’un booléen se fait dans le boucle c.à.d. ici test =1

for (n = 0; n < 100; n++)

{

test = 1;

for (i = 2; i < n; i++)

if (n%i == 0)

for (j = 2; j < i / 2; j++)

if (i%j == 0)

test = 0;

if (test)

printf("%d\t", n);

}

}

Exemple d’exécution :

**les premier 100 nombre speciaux sont :**

**0 1 2 3 4 5 6 7 8 9 10**

**14 15 17 19 21 22 23 25 26 29 31**

**35 37 38 39 41 43 46 47 49 51 53**

**58 59 61 62 65 67 69 71 73 74 77**

**83 85 86 87 89 91 93 94 95 97**

Exercise 2

#include <stdio.h>

void main()

{

int n;

float th, result;

printf("Donner le taux horaire de base : ");

scanf("%f", &th);

Saisir des donnes

printf("Donner le nombre d'heures de travail par semaine : ");

scanf("%d", &n);

Toujours tester lors de soustraction n dans le résultat donne un valeur positif grâce à l’utilisation des ‘if’.

if (n>44)

result = 39 \* 1 \* th + 5 \* 1.5 \* th + (n - 39 - 5) \* 2 \* th;

else

if (n>39)

result = 39 \* 1 \* th + (n - 39) \* 1.5 \* th;

else

result = n \* 1 \* th;

printf("Le salaire correspondant a %d heures de travail est : %0.0f $ ", n, result);

}

Exemple d’exécution :

**Donner le taux horaire de base : 6**

**Donner le nombre d'heures de travail par semaine : 47**

**Le salaire correspondant a 47 heures de travail est : 315 $**

Exercice 3

#include <stdio.h>

void main()

{

int n, i, sd = 0;

do

Saisir de n positive

On entre dans le boucle pour tout n < 0 si une fois n>0 on sort et continuer les instructions.

{

printf("Donner un entier positif : ");

scanf("%d", &n);

} while (n<0);

for (i = 1; i <= n; i++)

Calcul de somme des diviseurs de n et les mettre dans ‘sd’ qui est initialise par 0.

if (n%i == 0)

sd = sd + i; // ou sd+=i;

if (sd > 2 \* n)

printf("Ton nombre \"%d\" est abondant", n);

Affichage selon le cas.

else

printf("Ton nombre \"%d\" est non abondant", n);

}

Exercise 4

#include <stdio.h>

void main()

{

int n = 0;

double PA = 50000000, PB = 70000000;

while (PB>PA)

Calcul d’année qui commence par 0 et chaque année on a

PA = PA + 0.03 \* PA=PA (1+0.03) =PA\*1.03.

{

n++;

PA \*= 1.03;

PB \*= 1.02;

}

printf("Apres %d %d annees la population de A sera %lf,\n et la population de B sera %lf ", n, n, PA, PB);

}

Exemple d’exécution :

**Apres 35 annees la population de A sera 140693122.718576,**

**et la population de B sera 139992268.686372**