

Pratique du module 3 - Solutions

1.

[illegible]

```

    num = num + 1;
    cout << num << '\t' << num * num << '\t' << num * num * num <<
endl;
    num = num + 1;
    cout << num << '\t' << num * num << '\t' << num * num * num <<
endl;
    return 0;
}

```

2.

```

#include <iostream>
using namespace std;
main()
{
    int x, y, i, puissance;
    i = 1;
    puissance = 1;
    cin >> x;
    cin >> y;
    while (i <= y) {
        puissance *= x;
        ++i;
    }
    cout << puissance << endl;
    return 0;
}

```

3.

```

#include <iostream>
using namespace std;
#include <time.h>
main ()
{
    time_t temps_act;

    time(&temps_act); // Déterminer le temps actuel en secondes
    cout <<"La date et l'heure sont :"<< ctime(&temps_act);
    return 0;
}

```

4.

```
#include <iostream>
using namespace std;
#include <ctype.h>
#include <stdlib.h>

main()
{
    char lettre;
    do
    {
        cout <<"B Modification de l\'heure système "<<endl;

        cout <<"Q Quitter"<<endl;
        cout <<"Votre choix : "<<endl;
        cin >> lettre;
        if (lettre == 'B')
            system("TIME");
    }
    while (lettre != 'Q');
    return 0;
}
```

5.

```
#include <iostream>
#include <string.h>
using namespace std;
main()
{
    char tampon[64] = "J'aime programmer en C/C++";
    char destination[64];
    int longueur;
    longueur = strxfrm(destination, tampon, sizeof(tampon));
    cout <<"Longueur : "<< longueur <<endl;
    cout <<"Destination : " << destination <<endl;
    cout << "Tampon : " << tampon <<endl;
    return 0;
}
```

Note : Ce programme utilise la fonction `strxfrm` pour copier le contenu de la chaîne.

6.

```
#include <stdio.h>
#include <string.h>
using namespace std;

main()
{
    printf("Recherche de Sam dans Samuel %s\n",
           (strstr("Samuel", "Sam")) ? "Trouvé" : "Non trouvé");
    printf("Recherche de Jack dans Jacques %s\n",
           (strstr("Jacques", "Jack")) ? "Trouvé" : "Non trouvé");
    printf("Recherche de Christ dans Christine %s\n",
           (strstr("Christine", "Christ")) ? "Trouvé" : "Non trouvé");
}
```

7.

```
#include <stdio.h>
#include <ctype.h>
using namespace std;
main()
{
    char chaine[] = "J'aime bien programmer en C plus plus";
    int i;
    for (i = 0; chaine[i]; i++)
        putchar(toupper(chaine[i]));
    putchar('\n');
}
```

Note : La fonction `toupper` ne convertit que les minuscules; elle reste sans action sur tout autre caractère.

8.

```
#include <stdio.h>
#include <ctype.h>
using namespace std;
main()
{
    char chaine[] = "J'AIME BIEN PROGRAMMER EN C PLUS PLUS";
    int i;
    for (i = 0; chaine[i]; i++)
        putchar(tolower(chaine[i]));
    putchar('\n');
}
```

Note : La fonction `tolower` ne modifiera que les caractères réellement majuscules.

9.

```
#include <iostream>
using namespace std;
main()
{
    float litres, kilometres, totalLitres = 0.0, totalKilometres = 0.0, moyenne;
    cout << "Entrer le nombre de litres utilisés (-1 pour finir) : ";
    cin >> litres;
    while (litres != -1.0) {
        totalLitres += litres;
        cout << "Entrer le nombre de kilomètres parcourus : ";
        cin >> kilometres;
        totalKilometres += kilometres;
        cout << "Le nombre de km/litre pour ce plein est : " << kilometres/litres << endl << endl << "Entrer le nombre de litres utilisés (-1 pour finir) : ";
        cin >> litres;
    }
    moyenne = totalKilometres / totalLitres;
    cout << endl << "La moyenne globale de km/litre est : " << moyenne << endl;
    return 0;
}
```

10.

```
#include <iostream>
using namespace std;
main()
{
    int colonne = 8, ligne;
    while (colonne-- > 0) {
        ligne = 8;
        if (colonne % 2 == 0)
            cout << ' ';
        while (ligne-- > 0)
            cout << "* ";
        cout << endl;
    }
    return 0;
}
```

11.

```
#include <iostream>
using namespace std;
main()
{
    int somme = 0, nombre, valeur;
    cout << "Entrer le nombre de valeurs à additionner : ";
    cin >> nombre;
    for (int i = 1; i <= nombre; i++) {
        cout << "Entrer une valeur : ";
        cin >> valeur;
        somme += valeur;
    }
    cout << "La somme des " << nombre << " valeurs est : " << somme
    << endl;
    return 0;
}
```

12.

```
#include <iostream>
using namespace std;
#include <iomanip>

main()
{
    int produit, quantite;
    float total = 0.0;

    cout << "Entrer le numéro du produit et la quantité vendue." <<
    endl
    << "Entrer -1 pour indiquer la fin de l'entrée des données : ";
    cin >> produit;

    while (produit != -1) {
        cin >> quantite;
        switch (produit) {
            case 1:
                total += quantite * 2.98;
                break;
            case 2:
                total += quantite * 4.50;
                break;
            case 3:
                total += quantite * 9.98;
                break;
            case 4:
                total += quantite * 4.49;
                break;
        }
        cin >> produit;
    }
    cout << "Total : " << total << endl;
}
```

```

        break;
    case 5:
        total += quantite * 6.87;
        break;
    default:
        cout << " Numéro de produit invalide: " << produit <<
endl
        << "
        Quantité: " << quantite << endl;
        break;
    }

    cout << "Entrer le numéro du produit et la quantité vendue."
<< endl
<< " Entrer -1 pour indiquer la fin de l'entrée des données : ";
    cin >> produit;
}

cout.setf(ios::fixed | ios::showpoint);
cout << "La valeur totale des ventes est : " << setprecision(2)
<< total
<< endl;

return 0;
}

```

13.

```

#include <iostream>
using namespace std;

main()
{
// Moitié du haut
for (int ligne = 1; ligne <= 5; ligne++)
{
    for (int espace = 1; espace <= 5 - ligne; espace++)
        cout << ' ';
    for (int asterisque = 1; asterisque <= 2 * ligne - 1;
asterisque++)
        cout << '*';
    cout << endl;
}

// Moitié du bas
for (int ligne = 4; ligne >= 1; ligne--) {
    for (int espace = 1; espace <= 5 - ligne; espace++)
        cout << ' ';
    for (int asterisque = 1; asterisque <= 2 * ligne - 1;
asterisque++)
        cout << '*';
}
}

```

```

        cout << endl;
    }
    return 0;
}

```

14.

```

#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

void arrondir(void);

main()
{
    arrondir();
    return 0;
}

void arrondir(void)
{
    double x, y;

    cout.setf(ios::fixed | ios::showpoint);

    for (int boucle = 1; boucle <= 5; boucle++) {
        cout << "Entrer une valeur décimale : ";
        cin >> x;
        y = floor(x + .5);
        cout << x << " peut être arrondi à " << setprecision(1) << y
        << endl;
    }
}

```

15.

```

#include <iostream>
using namespace std;
int celsius(int);
int fahrenheit(int);
main()
{
    cout << "Degrés Fahrenheit équivalents aux températures en
    Celsius : "
    << endl << "Celsius\t\tFahrenheit" << endl;
    for (int i = 0; i <= 100; i++)
        cout << i << "\t\t" << fahrenheit(i) << endl;
    cout << endl << "Degrés Celsius équivalents aux températures en
    Fahrenheit : " << endl << "Fahrenheit\tCelsius" << endl;
}

```



```

    for (int i = 32; i <= 212; i++)
        cout << i << "\t\t" << celsius(i) << endl;
    return 0;
}

int celsius(int fTemp)
{
    return (int) (5.0 / 9.0 * (fTemp - 32));
}

int fahrenheit(int cTemp)
{
    return (int) (9.0 / 5.0 * cTemp + 32);
}

```

16.

```

#include <iostream>
using namespace std;
int cotePoints(int);
main()
{
    int moyenne;
    for (int boucle = 1; boucle <= 5; boucle++) {
        cout << endl << "Entrer la note moyenne de l'étudiant : ";
        cin >> moyenne;
        cout << moyenne << " vaut, sur une échelle de 4 points, "
        << cotePoints(moyenne) << endl;
    }
    return 0;
}

int cotePoints(int moyenne)
{
    if (moyenne >= 90)
        return 4;
    else if (moyenne >= 80)
        return 3;
    else if (moyenne >= 70)
        return 2;
    else if (moyenne >= 60)
        return 1;
    else
        return 0;
}

```

17.

```
#include <iostream>
#include <stdio.h>
using namespace std;
void salut(void)
{
    cout <<"Salut, tout le monde!" <<"\n";
}
void nombre_message (int nombre_fois)
{
    int compteur;
    for (compteur = 1; compteur <= nombre_fois; compteur++)
        salut();
}

main()
{
    cout <<"Affiche deux fois le message" <<"\n";
    nombre_message(2);
    cout <<"Affiche sept fois le message" <<"\n";
    nombre_message(7);

    return 0;
}
```

18.

```
#include <iostream>
using namespace std;
int TriplerParValeur (int);
void TriplerParReference (int &);

main()
{
    int valeur, &valeurRef = valeur;
    cout << "Entrer un entier : ";
    cin >> valeur;
    cout << endl << "La valeur avant l'appel de la fonction
TriplerParValeur () est : "
        << valeur << endl
        << "La valeur obtenue de TriplerParValeur () est : "<<
TriplerParValeur (valeur) << endl
        << " Après TriplerParValeur(), la valeur dans main est : "
<< valeur << endl << endl
        << "La valeur avant l'appel de TriplerParReference () est
: " << valeur << endl;

    TriplerParReference (valeurRef);
}
```

```

    cout << "La valeur (dans main) après l'appel de
TriplerParReference () est : "
        << valeur << endl;
    return 0;
}

```

19.

```

#include <iostream>
using namespace std;

int i_cube(int valeur)
{
    return(valeur * valeur * valeur);
}

main()
{
    cout <<"6 élevé au cube donne " << i_cube(6)<<endl;
    cout <<"9 élevé au cube donne " << i_cube(9)<<endl;
    cout <<"12 élevé au cube donne " << i_cube(12)<<endl;

    return 0;
}

```

20.

```

#include <iostream>
using namespace std;

main()
{
    int ligne, colonne;
    float table[3][5] = {{10.0, 20.0, 30.0, 40.0, 50.0},
                        {60.0, 70.0, 80.0, 90.0, 100.0},
                        {110.0, 120.0, 130.0, 140.0, 150.0}};

    for (ligne = 0; ligne < 3; ligne++)
        for (colonne = 0; colonne < 5; colonne++)
            cout <<" Ligne : " <<ligne << " Colonne : " << colonne <<
" "
            << "Nombre : " <<table[ligne][colonne]<<endl;

    return 0;
}

```

21.

```
#include <iostream>
#include <cstdlib>
using namespace std;

void tri_iteratif(int tableau[ ], int grosseur)
{
    int temp, i, j;

    for (i = 0; i < grosseur; i++)
        for (j = 0; j < grosseur; j++)
            if (tableau[i] < tableau[j])
            {
                temp = tableau[i];
                tableau[i] = tableau[j];
                tableau[j] = temp;
            }
}

main()
{
    int valeurs[35], i;

    for (i = 0; i < 35; i++)
        valeurs[i] = rand() % 1000;
    tri_iteratif(valeurs, 35);
    for (i = 0; i < 35; i++)
        cout << valeurs[i] << " ";

    return 0;
}
```

22.

```
#include <iostream>
#include <cstdlib>
using namespace std;

void tri_selectif(int tableau[ ], int grosseur)
{
    int temp, courant, j;

    for (courant = 0; courant < grosseur; courant++)
        for (j = courant + 1; j < grosseur; j++)
            if (tableau[courant] > tableau[j])
            {
                temp = tableau[courant];
                tableau[courant] = tableau[j];
            }
}
```

```

        tableau[j] = temp;
    }
}

main()
{
    int valeurs[30], i;

    for (i = 0; i < 30; i++)
        valeurs[i] = rand() % 1000;

    tri_selectif(valeurs, 30);

    for (i = 0; i < 30; i++)
        cout << valeurs[i]<< " " ;

    return 0;
}

```

23.

```

#include <iostream>
#include <string.h>
using namespace std;

void tri_iteratif(const char* tableau[], int taille)
{
    const char* temp;
    int i, j;

    for (i = 0; i < taille; i++)
        for (j = 0; j < taille; j++)
            if (strcmp(tableau[i], tableau[j]) < 0)
            {
                temp = tableau[i];
                tableau[i] = tableau[j];
                tableau[j] = temp;
            }
}

int main(void)
{
    const char* valeurs[]={"AA","CC","BB","EE","DD"};
    int i;

    tri_iteratif(valeurs, 5);

    for (i = 0; i < 5; i++)
        cout << valeurs[i] << " ";
}

```