Nom: VanHave Timothée Programmation 1 15 octobre 2021

4,1

1. (6 points) Déterminer si les chaînes de caractères suivantes sont des identificateurs valides en C++. Si la chaîne ne correspond pas à un identificateur valide, préciser pourquoi.

5

Chaîne	Valide	Problème
_bronze	oui	
Argent	oui	
or	Non	or est un mot clé réservé
2jambes	Non	un identificateur ne peut pas commencer par un chiffre
une_tête	Non	le caractère é est interdit pour un identificateur
main	Non	main est réservé

2. (18 points) Déterminer le type et la valeur des expressions suivantes. On suppose avoir déclaré :

11

```
int i = 5;
unsigned u = 4;
char c = 'C';
double d = 5.7, e = 2.2;
uint8_t f;
```

Expression	Type	Valeur
(short) e - (short) d	~~short~~	-3
(short) d - (unsigned) e	unsigned int	3
4/3*3.14	double	~~4/3,42~~
u - i < 0	bool	~~1~~
(char)(c-- + d)	char	~~G~~
i > u ? u : d	~~unsigned int~~	4
(int) e * ++d	double	13,4
7 > 6 ? [5 > 4 ? 3 : 2] : 1	int	3
f = 258 255. -256+3	~~int~~	~~258~~

9

3. (16 points) Qu'affichent les codes suivants ?

Code	Affichage

`cout << 011 << " "`
`<< 0b11 << " "`
`<< 0x11 << " "`
`<< 110 ;`
	9 3 17 110

`int i = 2;`
`{`
`  float i = 2.2`
`  cout << i;`
`  {`
`  char i = 'i'`
`  cout << i;`
`  }`
`  cout << i;`
`}`
`cout << i;`
	2.2 i 2.2 2

`int a = 0;`
`if (a =! 2)`
`  cout << "a different de 2";`
`else`
`  cout << "a vaut 2: " << a;`
	a ~~different~~ de 2

`cout << setprecision(4) << 3.14159;`	~~3.1415e+0~~
`cout << fixed << setprecision(4) << 3.14159;`	~~3.1415~~
`cout << boolalpha << (2 < 2 == 0) ;`	true

`int i = 13, j = 0;`
`while (i > j++){`
`  i /= 2;}j++;`
`cout << i << " " << j;`
	~~6~~ ~~15~~

`int i = -2, j = 3;`
`do {`
`  j--;`
`} while (j += i++);`
`cout << i << " " << j;`
	~~0~~ ~~-1~~

4. (12 points) Écrire une expression retournant la valeur `true` si, et seulement si la proposition est évaluable et vraie et `false` sinon. On suppose avoir déclaré :

5

```
int a, b, c;
char e;
```

Proposition	Expression de type `bool`
`b` est différent de 0	b!=0
`a` est plus grand que `b`	a>b
La suite des valeurs de `a`, `b` et `c` est strictement croissante	a<b and b>c
`a/b` est plus grand que `c/b`	b!=0 and a/b>c/b
L'addition de `a` et de `b` ne provoquera pas un débordement du type `int`	
Le caractère contenu dans `e` est un chiffre	e>19 and e<31
`a` est pair	!(a%2)

5. (10 points) Écrire un programme complet qui permet d'afficher le plus petit diviseur supérieur à 1 d'un nombre entier plus grand que 1 introduit par l'utilisateur. On suppose que l'utilisateur donne toujours une valeur valide ; pas besoin de traiter les entrées erronées.

9

```
#include<iostream>
using namespace std;
int main(){
int number=0, cnt=2;
cout<< "Entrez un nombre entier" << endl;
cin>>number
while(number%cnt!=0)
    cnt++;
cout<<cnt<<"est le plus petit diviseur de "<<number;
return 1;
}
```

2

6. (10 points) Pour calculer le nombre de possibilités d'obtenir un montant de 100 avec des pièces de 20, 10 et 5, une personne propose le programme suivant :

```
#include <iostream>
using namespace std;
int main ()
{
  int cpt = 0;
  for (int n20 = 0; n20 <= 5; n20++)
    for (int n10 = 0; n10 <= 10; n10++)
      for (int n5 = 0; n5 <= 20; n5++)
        if (5 * n5 + 10 * n10 + 20 * n20  == 100)
          cpt++;
  cout << "Il y a " << cpt << " possibilites\n";
}
```

Ce programme est maladroit pour plusieurs raisons. Tout d'abord, la dernière boucle `for` est inutile, car il n'y a qu'une possibilité de compléter la somme avec des pièces de 5 pour arriver à 100 (si on ne dépasse pas déjà la somme avec les autres pièces). Ensuite, si l'utilisateur voulait connaître le nombre de possibilités pour arriver à un autre montant, il faudrait modifier plusieurs lignes du programme et le compiler à nouveau.

Modifier le programme de telle sorte que :

— L'utilisateur puisse spécifier un montant quelconque réalisable avec des pièces de 5, 10 et 20 (on suppose que l'utilisateur donne toujours un montant valide ; pas besoin de traiter les entrées erronées).

— Le programme ne comporte pas de `if`

```
#include <iostream>
using namespace std;
int main(){
  int montant=0, cpt=0;
  cout << "veuillez entrer un montant" << endl;
  cin >> montant;

  for(int n20=0; n20< 5; n20++)
     for(int n10=0; n10< 10; n10++){
        (n20*2 + n10*10)% 100 != 0 and cpt += montant-(n20+n10)/5;
     }
  cout << "il y a " << cpt << "possibilits";
  return 1;
}
```