



## Practicum woensdag en vrijdag: van Python naar C++

Collapse context

Your answer passed the tests! Your score is 100.0%. [Submission #6733cac06c698110d4c19d1d (2024-11-12 21:38:08)]



### Question 1: Basis loop

✓ Perfect



Schrijf een functie `computeSom` die alle getallen van een gegeven vector optelt en het resultaat teruggeeft.

```
1 int computeSom(vector<int> values){
2     // hier komt je code
3     int result = 0;
4     for (int i:values)
5     {
6         result += i;
7     }
8     return result;
9 }
```

Submit

### Question 2: Verschillende loops (a)

✓ Perfect



Schrijf een functie `printNames` die een vector van strings met namen op het scherm afdrukt. Zet elke naam op een nieuwe regel. Test het uit door een vector aan te maken waarin deze namen zijn opgeslagen: "Toon", "Tom", "Wannes", "Daphne". Loop vervolgens *met een index* over deze vector en print elk element van de vector op een nieuwe regel. Merk op dat een functie die niets teruggeeft het type `void` teruggeeft. Vergeet niet om elke cout te laten eindigen met een `<< endl`. Dat is enkel belangrijk voor INGINIOUS.

```
1 void printNames(vector<string> names){
2     for(int i = 0; i < names.size(); i++)
3     {
4         cout << names[i] << endl;
5     }
6 }
7 }
```

Submit

### Question 3: Verschillende loops (b)

✓ Perfect



Kopieer en plak je code van de vorige vraag hieronder. Loop opnieuw door de vector en print alle namen af. Maak dit keer gebruik van *de korte notatie* (:) om door de vector te lopen.

```
1 void printNames(vector<string> names){
2     for (string name:names){
3         cout << name << endl;
4     }
5
6 }
```

Submit

#### Question 4: Verschillende loops (c)

✓ Perfect

Schrijf een functie `removeFirstAndPrint` die het eerste element van de vector verwijdert en vervolgens alle (overblijvende) namen afdruckt op het scherm. Maak dit keer gebruik *van een iterator* om door de vector te lopen.

```
1 void removeFirstAndPrint(vector<string> names){
2     // hier komt je code
3     names.erase(names.begin());
4     for (string i:names){
5         cout << i << endl;
6     }
7 }
```

Submit

#### Question 5: Maps (a)

✓ Perfect

Schrijf een functie `printContacts` die een map van int en strings krijgt, met daarin namen en de daarbijhorende telefoonnummers: bv (123, Toon), (456, Tom) en (789, Daphne). Loop *met een iterator* over de map en print elke naam met elk telefoonnummer in het volgende formaat: `Toon: 123`

```
1 void printContacts(map<int,string> contacts){
2     // hier komt je code
3     for(auto i = contacts.begin(); i != contacts.end(); ++i)
4     {
5         cout << i->second << ": " << i->first << endl;
6     }
7 }
```

Submit

#### Question 6: Maps (b)

✓ Perfect

Schrijf een functie `removeAndPrint` die een map van int en strings krijgt zoals in de vorige vraag en een telefoonnummer. Maak gebruik van het telefoonnummer om iemand uit de map te verwijderen. Print vervolgens de overgebleven namen en telefoonnummers. Maak dit keer gebruik van *de korte notatie* (`:`) om door de map te lopen. Je geeft het telefoonnummer van de te verwijderen contactpersoon mee als

tweede parameter.

```
1 void removeAndPrint(map<int,string> contacts, int telToBeRemoved){
2     // hier komt je code
3     int times = 0;
4     contacts.erase(telToBeRemoved);
5     for (auto& c : contacts)
6     {
7         cout << c.second << ": " << c.first << endl;
8     }
9 }
```

Submit

### Question 7: Frequentietabel opstellen

✓ Perfect

Schrijf een functie `createFreqTab` die vertrekt van een vector van strings en een frequentietabel teruggeeft (telkens de string als key en het aantal keer dat die string voorkomt als waarde). Gebruik hiervoor een map.

Voor de vector {"dit","is","een","dit","een","dit","hallo"} wordt dit de map {"dit", 3}, {"is", 1}, {"een", 2}, {"hallo", 1};

```
dit 3
is 1
een 2
hallo 1
```

```
1 map<string,int> createFreqTab(vector<string> woorden){
2     map<string,int> result;
3     for (auto w : woorden)
4         result[w] = 0;
5
6     for (auto w : woorden)
7         result[w]++;
8
9     return result;
10 }
```

Submit

### Question 8: Frequentietabel afdrukken

✓ Perfect

Schrijf een functie `printFreqTab` die een frequentietabel als volgt op het scherm afdrukt:

```
dit 3
is 1
een 2
hallo 1
```

De code uit de vorige vraag wordt automatisch ingevoegd. Bij de testcode wordt er eerst een frequentietabel opgesteld met de code uit de vorige vraag en vervolgens afgedrukt met de code uit deze vraag.

```
1 void printFreqTab(map<string,int> freqtab){
2     for (auto pair : freqtab)
```

```
3     {
4         cout << pair.first << ' ' << pair.second << endl;
5     }
6 }
```

Submit

### Question 9: Vectoren (a)

✓ Perfect

✕

We gaan nu met (wiskundige) vectoren in c++ werken! Maak een functie `vecToString(vector<int> v1)` die een vector van integers als input krijgt (bijv. `vector<int> v1 = {1, 2, 3, 4, 5};`) en een string van deze vector als output geeft ("`[1, 2, 3, 4, 5]`"). Gebruik de functie `to_string()` om een integer om te zetten naar een string.

```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4
5 using namespace std;
6
7 string vecToString(vector<int> v1)
8 {
9     string result = "[";
10    for (int &i : v1)
11    {
12        if (&i != &v1.back())
13            result += to_string(i) + ", ";
14        else
15            result += to_string(i) + ']';
16    }
17    return result;
18 }
```

Submit

### Question 10: Vectoren (b)

✓ Perfect

✕

Maak een functie `addVector(vector<int> v1, int x)`. In deze functie wordt een integer x bij elk element van de vector v1 opgeteld. De functie geeft een vector terug. `addVector({1,2,3},4)` geeft `{5,6,7}` terug. De code van de vorige vraag wordt gebruikt om na te kijken of het correct is.

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 vector<int> addVector(vector<int> v1, int x){
6     for (int& i : v1)
7         i+=x;
8
9     return v1;
10 }
```

Submit

### Question 11: Vectoren (c)

✓ Perfect

✕

Schrijf een functie `addVectors(vector<int> v1, vector<int> v2)`, die de elementsgewijze som van twee integer vectoren teruggeeft. De functie geeft een vector terug. `addVectors({1,2,3},{0,-1,4})` geeft de vector `{1,1,7}` terug.

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 vector<int> addVectors(vector<int> v1, vector<int> v2){
6     vector<int> result;
7     for (int i = 0, n = v1.size(); i < n; i++)
8     {
9         result.push_back(v1[i] + v2[i]);
10    }
11    return result;
12 }
```

Submit

### Question 12: Vectoren (d)

✓ Perfect

✕

Maak tenslotte een functie `normVector(vector<int> v1)` die de norm van een vector teruggeeft. De norm van een vector is de wortel van de som van de kwadraten van alle elementen van de vector. De norm van (3,0,4) is de wortel van  $9+0+16$  wat gelijk is aan 5. Maak hierbij gebruik van de functies `sqrt` en `pow` (van power, om machten te berekenen) van `cmath`.

```
1 #include <iostream>
2 #include <vector>
3 #include <cmath>
4 using namespace std;
5
6 float normVector(vector<int> v1){
7     float result=0.f;
8
9     for (int i : v1)
10         result += i * i;
11
12     return sqrt(result);
13 }
```

Submit

### Question 13: Print Text Rectangle (a)

✓ Perfect

✕

In deze opdracht ga je een functie `printTextRectangleFilled(int width, int height)` maken. Gegeven de hoogte en een breedte print deze functie een rechthoek van asterisk \* symbolen. Bijv. `printTextRectangleFilled(4, 3)`

```
****
****
****
```

```
1 #include <iostream>
2 using namespace std;
3
4 void printTextRectangleFilled(int width, int height)
5 {
6     for (int i = 0; i < height; i++)
7     {
8         for (int j = 0; j < width; j++)
9         {
10             cout << '*';
11         }
12         cout << endl;
13     }
14 }
```

Submit

#### Question 14: Print Text Rectangle (b)

✓ Perfect



In deze opdracht ga je een functie `printTextRectangleOutline(int width, int height)` maken. Gegeven de hoogte en een breedte van een rechthoek print deze functie de omtrek van een rechthoek van asterisk \* symbolen. Bijv. `printTextRectangleOutline(4, 3)`

```
****
*  *
****
```

```
1 #include <iostream>
2 using namespace std;
3
4 void printTextRectangleOutline(int width, int height)
5 {
6     for (int i = 0; i < height; i++)
7     {
8         if (i == 0 or i == height - 1)
9         {
10             for (int j = 0; j < width; j++)
11                 cout << '*';
12             cout << endl;
13             continue;
14         }
15         cout << '*';
16         for (int j = 0; j < width - 2; j++)
17             cout << ' ';
18         cout << '* ' << endl;
19     }
20 }
```

Submit

### Question 15: Priemgetal

✓ Perfect

✕

Maak een functie `isPrime(int n)` die `true` als teruggeeft als `n` een priemgetal is, en `false` als output geeft als `n` geen priemgetal is. Maak hierbij gebruik van een while-loop.

```
1  #include <iostream>
2  using namespace std;
3
4  bool isPrime(int n)
5  {
6      if (n == 1)
7          return false;
8
9      for (int i = 3; i < n; i+=2)
10     {
11         if (!(n % 2) || !(n % i))
12             return false;
13     }
14     return true;
15 }
```

Submit

### Question 16: Efficient Priemgetal

✓ Perfect

✕

In deze opdracht ga je een meer efficiënte functie `isPrime(int n)` maken. Vertaal hiervoor onderstaande pseudocode naar een C++ functie.

```
1  /* Pseudocode:
2  function is_prime(n)
3      if n <= 1
4          return false
5      else if n <= 3
6          return true
7      else if n mod 2=0 or n mod 3=0
8          return false
9      i = 5
10     while i * i <= n
11         if n mod i = 0 or n mod (i + 2) = 0
12             return false
13         i = i + 6
14     return true
15 */
16 bool isPrime(int n)
17 {
18     if (n <= 1)
19         return false;
20     else if (n <= 3)
21         return true;
22     else if (n % 2 == 0 || n % 3 == 0)
23         return false;
24     int i = 5;
25     while (i * i <= n)
```

```

26     {
27         if (n % i == 0 || n % (i + 2) == 0)
28             return false;
29         i += 6;
30     }
31     return true;
32 }
33

```

Submit

### Question 17: Matrix - Upper triangle

✓ Perfect



Een matrix stel je in C++ voor door een vector van vectoren (net zoals je in Python een lijst van lijsten zou gebruiken).

Maak een functie `upper_triangle` die een matrix als input krijgt en de *upper triangle* van de matrix als output geeft. Alles onder de hoofddiagonaal van de matrix wordt op 0 gezet. Als deze functie matrixA als input zou krijgen zou matrixB de output van de functie zijn.

```

vector<vector<int>> matrixA = {{1,2,3},{4,5,6},{7,8,9}};

vector<vector<int>> matrixB = {{1,2,3},{0,5,6},{0,0,9}};

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  vector<vector<int>> upper_triangle(vector<vector<int>> &m)
6  {
7      for (int i = 0; i < m.size(); i++)
8          for (int j = 0; j < i; j++)
9              m[i][j] = 0;
10
11     return m;
12 }
13

```

Submit