

Status	Finished
Started	Monday, 23 December 2024, 7:07 PM
Completed	Monday, 23 December 2024, 7:38 PM
Duration	30 mins 1 sec
Marks	90.00/90.00
Grade	10.00 out of 10.00 (100%)

## Question 1

Correct

Mark 10.00 out of 10.00

**[Template Statistics]**

Viết [mẫu hàm](#) `void printStatistics(T a[], int n)` để thực hiện một [thống kê](#) cơ bản về số lớn nhất, số nhỏ nhất của dãy số  $a$ . Tham số đầu vào của hàm là mảng  $a$  chứa các số có kiểu  $T$  và  $n$  là số lượng phần tử của mảng đó. Hàm `printStatistics()` có nhiệm vụ in ra số lớn nhất và nhỏ nhất của mảng  $a$ .

**Đầu vào**

Gồm hai dòng được nhập vào từ bàn phím:

- Dòng thứ nhất là số  $n$  là số lượng phần tử của dãy  $a$ .
- Dòng thứ hai chứa  $n$  phần tử của dãy  $a$ , các phần tử cách nhau bởi một dấu cách.

**Đầu ra**

In ra màn hình số lớn nhất và nhỏ nhất của dãy  $a$  trên một dòng. Hai số cách nhau bởi một khoảng trống.

**Lưu ý:** Bạn chỉ cần hoàn thành hàm `printStatistics` theo yêu cầu. Phần nhập dữ liệu đầu vào đã được lập trình sẵn.

**For example:**

Input	Result
5	5 1
1 2 3 4 5	3.3 1.2
4	
1.2 3.3 2.2 2.8	

**Answer:** (penalty regime: 0 %)

```
1 | template<class T>
2 | void printStatistics(T a[], int n)
3 | {
4 |     cout << *max_element(a, a + n) << " " << *min_element(a, a + n) << endl;
5 | }
```

	Input	Expected	Got	
✓	5 1 2 3 4 5 4 1.2 3.3 2.2 2.8	5 1 3.3 1.2	5 1 3.3 1.2	✓

Passed all tests! ✓

► SHOW/HIDE QUESTION AUTHOR'S SOLUTION (CPP)

Correct

Marks for this submission: 10.00/10.00.

## Question 2

Correct

Mark 10.00 out of 10.00

**[Template Max Of Two]**

Viết [mẫu hàm](#) `T getMax(T a, T b)` nhận tham số là hai biến số `a`, `b` thuộc kiểu `T` và trả về giá trị lớn hơn trong hai giá trị `a`, `b`.

**Đầu vào**

Một dòng duy nhất từ bàn phím chứa hai số `a` và `b` cách nhau bởi một khoảng trống.

**Đầu ra**

In ra màn hình một dòng duy nhất chứa [số lớn hơn](#) trong hai số `a` và `b`.

**Lưu ý:** Bạn chỉ cần hoàn thành hàm `getMax` trả về giá trị lớn hơn của hai biến. Phần nhập dữ liệu và in kết quả ra màn hình đã được lập trình sẵn.

**For example:**

Input	Result
3 4	4
6.2 5.2	6.2

**Answer:** (penalty regime: 0 %)

```
1 | template<class T>
2 | T getMax(T a, T b)
3 | {
4 |     return max(a, b);
5 | }
6 |
```

	Input	Expected	Got	
✓	3 4 6.2 5.2	4 6.2	4 6.2	✓
✓	3 1 10.2 3.7	3 10.2	3 10.2	✓
✓	4 7 21.0 17.1 6 6 124.2 225.5	7 21 6 225.5	7 21 6 225.5	✓

Passed all tests! ✓

► **SHOW/HIDE QUESTION AUTHOR'S SOLUTION (CPP)**

Correct

Marks for this submission: 10.00/10.00.

## Question 3

Correct

Mark 10.00 out of 10.00

**[Get Sum Of Two]**

Viết [mẫu hàm](#) `T getSum(T a, T b)` nhận tham số đầu vào là hai biến `a, b` thuộc kiểu số `T` và trả về tổng của `a` và `b`.

**Đầu vào**

Một dòng duy nhất từ bàn phím chứa hai số `a` và `b` cách nhau bởi một khoảng trống.

**Đầu ra**

Một dòng duy nhất chứa tổng của hai số `a` và `b`.

**Lưu ý:** Bạn chỉ cần hoàn thành hàm `getSum` để tính tổng của hai số. Phần nhập dữ liệu đầu vào và in kết quả ra màn hình đã được lập trình sẵn.

**For example:**

Test	Input	Result
float a, b;	3 4	7
while (cin >> a >> b) { cout << getSum(a, b) << endl; }	6.2 5.2	11.4

**Answer:** (penalty regime: 0 %)

```
1 | template <class T>
2 | T getSum(T a, T b)
3 | {
4 |     return a + b;
5 | }
```

	Test	Input	Expected	Got	
✓	<pre>float a, b; while (cin &gt;&gt; a &gt;&gt; b) {     cout &lt;&lt; getSum(a, b) &lt;&lt; endl; }</pre>	3 4 6.2 5.2	7 11.4	7 11.4	✓
✓	<pre>float a, b; while (cin &gt;&gt; a &gt;&gt; b) {     cout &lt;&lt; getSum(a, b) &lt;&lt; endl; }</pre>	3 1 10.2 3.7	4 13.9	4 13.9	✓

Passed all tests! ✓

► SHOW/HIDE QUESTION AUTHOR'S SOLUTION (CPP)

Correct

Marks for this submission: 10.00/10.00.

## Question 4

Correct

Mark 10.00 out of 10.00

**[Get Sum Of An Array]**

Viết [mẫu hàm](#) `T getSum(T a[], int n)` để tính tổng của dãy số  $a$ . Tham số đầu vào của hàm là mảng  $a$  chứa các phần tử kiểu số  $T$  và biến  $n$  là số phần tử của mảng  $a$ . Nhiệm vụ của hàm `getSum` là trả về tổng các phần tử của mảng  $a$ .

**Đầu vào**

Gồm hai dòng nhập vào từ bàn phím:

- Dòng thứ nhất chứa số  $n$  là số lượng phần tử của dãy.
- Dòng thứ hai chứa  $n$  phần tử của dãy cách nhau bởi một khoảng trống.

**Đầu ra**

In ra màn hình một dòng duy nhất chứa tổng các phần tử của dãy.

**Lưu ý:** Bạn chỉ cần hoàn thành hàm `getSum` trả về tổng giá trị của các phần tử trong dãy. Phần nhập dữ liệu và in kết quả ra màn hình đã được lập trình sẵn.

**Answer:** (penalty regime: 0 %)

```
1 | template<class T>
2 | T getSum(T a[], int n)
3 | {
4 |     T ans = 0;
5 |     for(int i = 0; i < n; i++) ans += a[i];
6 |     return ans;
7 | }
```



	Input	Expected	Got	
✓	5 1 2 3 4 5 4 1.2 3.3 2.2 2.8	15 9.5	15 9.5	✓
✓	6 1 2 3 4 5 3 5 1.2 3.3 2.2 2.8 0.2	18 9.7	18 9.7	✓

Passed all tests! ✓

► **SHOW/HIDE QUESTION AUTHOR'S SOLUTION (CPP)**

Correct

Marks for this submission: 10.00/10.00.

## Question 5

Correct

Mark 10.00 out of 10.00

**[Output Function]**

Cho nguyên [mẫu hàm](#) sau để hiển thị số nguyên ra màn hình: `void output(const int &);`

Các bạn hãy viết định nghĩa hàm trên (chỉ cần định nghĩa hàm này) mà không cần viết hàm `main()`, thư viện cũng đã có sẵn cho bạn.

**For example:**

Input	Result
1	1

**Answer:** (penalty regime: 0 %)

```
1 | template<class T>
2 | void output(const T &x)
3 | {
4 |     cout << x;
5 | }
```

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

## Question 6

Correct

Mark 10.00 out of 10.00

**[Check Prime Number]**

Cho nguyên mẫu hàm sau để kiểm tra một số có phải là số nguyên tố không: `bool isPrime(int);`

Hàm `isPrime` sẽ trả về giá trị **true** nếu n là số nguyên tố và giá trị **false** trong trường hợp ngược lại. Các bạn hãy viết định nghĩa hàm trên (chỉ cần định nghĩa hàm này) mà không cần viết `main()`, thư viện cũng có sẵn cho bạn rồi.

**Answer:** (penalty regime: 0 %)

```
1 | template<class T>
2 | T isPrime(T x)
3 | {
4 |     if(x < 2) return false;
5 |     for(T i = 2; i < sqrt(x); i++)
6 |     {
7 |         if(x % i == 0) return false;
8 |     }
9 |     return true;
10| }
```

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

## Question 7

Correct

Mark 10.00 out of 10.00

**[HappyNumber]**

**Số may mắn** là số được định nghĩa theo quá trình sau: bắt đầu với số nguyên dương  $x$  và tính tổng bình phương  $y$  các chữ số của  $x$ , sau đó tiếp tục tính tổng bình phương các chữ số của  $y$ . Quá trình này lặp đi lặp lại cho đến khi thu được kết quả là 1 thì dừng (tổng bình phương các chữ số của số 1 chính là 1) hoặc quá trình sẽ kéo dài vô tận. Số mà quá trình tính này kết thúc bằng 1 gọi là số may mắn. Số có quá trình tính kéo dài vô tận là **số không may mắn** hay còn gọi là **số đen đui**.

Ví dụ, 7 là số may mắn:

$$0^2 + 7^2 = 494^2 + 9^2 = 979^2 + 7^2 = 1301^2 + 3^2 + 0^2 = 101^2 + 0^2 = 1$$

Viết hàm `bool isHappyNumber(int n)` trả về `true` nếu số  $n$  là số may mắn ngược lại trả về `false`.

**For example:**

Test	Input	Result
<pre>int n; cin &gt;&gt; n; cout &lt;&lt; isHappyNumber(n);</pre>	19	1

**Answer:** (penalty regime: 0 %)

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  #define el "\n"
4  #define ll long long
5  #define ull unsigned long long
6  #define se second
7  #define fi first
8  #define be begin()
9  #define en end()
10 #define Faster cin.tie(0); cout.tie(0); ios_base::sync_with_stdio(0);
11
12 int solve(int n)
13 {
14     int sum = 0;
15     while(n)
16     {
17         sum += pow(n % 10, 2);
18         n /= 10;
19     }
20     return sum;
21 }
22
23 bool isHappyNumber(int n)
24 {
25     /*while(n < 10)
26     {
27         n = n * n;
28     }
29     while(n >= 10)
30     {
31         n = solve(n);
32     }*/
33     unordered_map<int, int> mp;
34     while(n != 0 && mp[n] == 0)
35     {
36         mp[n]++;
37         n = solve(n);
38     }
39     if(n == 1) return true;
40     return false;
```

41 | }

	Test	Input	Expected	Got	
✓	int n; cin >> n; cout << isHappyNumber(n);	19	1	1	✓
✓	int n; cin >> n; cout << isHappyNumber(n);	18	0	0	✓
✓	int n; cin >> n; cout << isHappyNumber(n);	365	1	1	✓
✓	int n; cin >> n; cout << isHappyNumber(n);	487	1	1	✓
✓	int n; cin >> n; cout << isHappyNumber(n);	488	0	0	✓
✓	int n; cin >> n; cout << isHappyNumber(n);	7	1	1	✓
✓	int n; cin >> n; cout << isHappyNumber(n);	70	1	1	✓
✓	int n; cin >> n; cout << isHappyNumber(n);	100	1	1	✓
✓	int n; cin >> n; cout << isHappyNumber(n);	8	0	0	✓
✓	int n; cin >> n; cout << isHappyNumber(n);	999	0	0	✓

Passed all tests! ✓

► SHOW/HIDE QUESTION AUTHOR'S SOLUTION (CPP)

Correct

Marks for this submission: 10.00/10.00.

## Question 8

Correct

Mark 10.00 out of 10.00

**[LongestHarmoniousSubsequence]**

Dãy hài hòa là một dãy thỏa mãn điều kiện giá trị lớn nhất và giá trị nhỏ nhất của dãy khác nhau đúng bằng  $\lfloor \frac{1}{2} \rfloor$ .

Viết hàm `int findLHS(int arr[], int n)` trả về độ dài của dãy hài hòa con dài nhất có thể có trong mảng giá trị nguyên `arr` có `n` phần tử

Ví dụ:

Input: 1 3 2 2 5 2 3 7

Output: 5 - Dãy hài hòa con dài nhất là [3,2,2,2,3]

For example:

Test	Input	Result
findLHS(arr, n)	4 1 2 3 4	2

Answer: (penalty regime: 0 %)

```

1  #include<bits/stdc++.h>
2  int findLHS(int arr[], int n)
3  {
4      vector<int> vt;
5      unordered_map<int,int> mp;
6      for(int i = 0; i < n; i++)
7      {
8          if(!mp[arr[i]]) vt.push_back(arr[i]);
9          mp[arr[i]] ++;
10     }
11     sort(vt.begin(), vt.end());
12     int Size = vt.size();
13     int ans = mp[vt[0]];
14     for(int i = 1; i < Size; i++)
15     {
16         if(vt[i] - vt[i - 1] <= 1) ans = max(ans, mp[vt[i]] + mp[vt[i-1]]);
17     }
18     return ans;
19 }
```

	Test	Input	Expected	Got	
✓	findLHS(arr, n)	4 1 2 3 4	2	2	✓
✓	findLHS(arr, n)	25 27 20 17 32 7 49 19 52 26 1 16 4 47 42 49 47 35 51 4 8 40 15 45 39 12	2	2	✓
✓	findLHS(arr, n)	50 15 30 27 17 0 31 37 42 47 18 22 37 48 0 41 23 1 1 5 14 36 41 18 46 29 21 26 35 27 46 44 22 22 43 38 2 46 0 15 22 8 24 12 34 21 37 11 39 17 48	6	6	✓
✓	findLHS(arr, n)	150 6 26 15 84 26 99 17 99 2 14 63 56 97 61 57 0 4 15 82 60 96 47 24 46 36 25 49 54 4 55 62 89 70 12 49 36 85 93 29 96 17 70 36 52 83 44 0 7 23 50 44 36 42 97 42 28 44 17 21 18 16 66 69 40 42 17 91 3 2 95 23 21 79 23 4 8 63 72 96 58 68 86 79 36 79 93 30 51 87 29 11 58 48 57 52 68 20 43 63 65 92 10 16 7 61 64 52 27 50 59 46 1 19 65 42 11 85 25 4 14 78 13 38 46 20 86 47 22 39 86 69 35 79 74 88 85 12 93 88 71 77 52 94 66 29 27 25 93 99 49	6	6	✓
✓	findLHS(arr, n)	250 37 2 44 4 6 40 9 3 31 18 8 13 12 16 18 34 16 44 35 42 46 42 26 12 33 5 46 6 35 14 51 9 53 24 17 29 29 37 49 10 52 38 50 46 8 27 26 19 29 35 6 36 30 19 16 28 37 41 53 27 4 18 35 4 43 33 29 46 41 15 22 50 18 30 0 0 7 42 10 42 36 41 29 6 47 30 3 24 44 25 22 52 29 1 26 19 48 33 39 51 33 7 9 47 30 18 27 21 46 2 51 37 19 44 51 51 13 8 50 4 26 28 15 28 22 34 32 31 12 10 10 47 2 23 21 46 18 5 54 21 48 40 18 4 21 2 43 36 3 54 31 52 24 49 37 30 45 32 15 20 4 32 25 28 6 5 36 43 21 54 19 33 17 1 10 25 12 51 33 30 18 44 14 50 1 11 35 43 31 35 15 52 39 36 53 13 36 49 0 39 39 31 34 4 44 5 17 29 12 39 8 43 39 49 43 16 10 24 14 54 42 54 0 1 34 50 41 29 47 23 39 35 14 41 21 5 41 50 5 4 33 1 2 35 13 46 5 10 1 25	15	15	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 9

Correct

Mark 10.00 out of 10.00

[Template Swap ]

Viết [mẫu hàm](#) `void swapNumber(T & x, T & y)` thực hiện việc hoán đổi giá trị của hai biến  $\backslash(x,y\backslash)$  là hai số có cùng kiểu  $\backslash(T)\backslash$ .

For example:

Input	Result
3 4	4 3
6.2 5.2	5.2 6.2

Answer: (penalty regime: 0 %)

```
1 | template<class T>
2 | void swapNumber(T &x, T &y)
3 | {
4 |     swap(x, y);
5 | }
```

	Input	Expected	Got	
✓	3 4	4 3	4 3	✓
	6.2 5.2	5.2 6.2	5.2 6.2	
✓	3 1	1 3	1 3	✓
	10.2 3.7	3.7 10.2	3.7 10.2	

Passed all tests! ✓



Correct

Marks for this submission: 10.00/10.00.

[Back to Course](#)