

Status	Finished
Started	Tuesday, 24 December 2024, 10:28 AM
Completed	Tuesday, 24 December 2024, 2:23 PM
Duration	3 hours 55 mins
Marks	100.00/100.00
Grade	10.00 out of 10.00 (100%)

Question 1

Correct

Mark 10.00 out of 10.00

[MaxAndMinFromFile]

Viết chương trình nhận đầu vào là một xâu kí tự chứa đường dẫn đến một tệp. Thử mở tệp này, nếu không thành công in ra "Mission failed". Trong trường hợp mở được, tệp sẽ chứa các số nguyên cách nhau bởi dấu cách hoặc/và dấu xuống dòng. In ra màn hình số lớn nhất và nhỏ nhất trong tệp cách nhau bởi dấu cách.

For example:

Input	Result
1.txt	23 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 int main()
12 {
13     Faster;
14     string path;
15     getline(cin, path);
16     ifstream file(path);
17     if(file.is_open())
18     {
19         int num;
20         vector<int> vt;
21         while(file >> num)
22         {
23             vt.push_back(num);
24         }
25         sort(vt.begin(), vt.end());
26         cout << vt.back() << " " << vt[0];
27     }
28     else
29     {
30         cout << "Mission failed";
31     }
32     return 0;
33 }
34

```

	Input	Expected	Got	
✓	1.txt	23 1	23 1	✓
✓	2.txt	58 1	58 1	✓
✓	3.txt	65 1	65 1	✓
✓	4.txt	Mission failed	Mission failed	✓

	Input	Expected	Got	
✓	5.txt	89475641 5	89475641 5	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 2

Correct

Mark 10.00 out of 10.00

[Perimeter]

Viết chương trình, nhập vào đường dẫn đến một tệp. Tệp này chứa thông tin của các hình vuông, hình tròn và [hình chữ nhật](#). In ra chu vi của các hình trong tệp trên từng dòng, theo thứ tự xuất hiện trong tệp. Kết quả được làm tròn đến hai chữ số thập phân sau dấu phẩy.

Một tệp ví dụ như sau:

```
TYPE: SQUARE
SIDE: 5.3

TYPE: RECTANGLE
WIDTH: 5.0 HEIGHT: 16.0

TYPE: CIRCLE
RADIUS: 2.15
```

Một tệp có thể chứa thông tin của một hoặc nhiều hình và không theo thứ tự nhất định. Mỗi hình sẽ được định nghĩa trên hai dòng, dòng đầu tiên là loại hình (SQUARE, RECTANGLE, CIRCLE) và dòng tiếp theo là các thông số tương ứng với hình đó (SIDE, WIDTH, HEIGHT, RADIUS). Giữa thông tin của hai hình sẽ có một dòng trống.

For example:

Input	Result
1.txt	28.09
	80.00
	13.51

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 int main()
12 {
13     Faster;
14     string path;
15     getline(cin, path);
16     ifstream file(path);
17     string s;
18     while(file >> s)
19     {
20         string type;
21         file >> type;
22         if(type == "CIRCLE")
23         {
24             double rad;
25             file >> s >> rad;
26             cout << fixed << setprecision(2) << 2 * rad * 3.1415926535897 << el;
27         }
28         else if(type == "RECTANGLE")
```

```

29  {
30      double wi, he;
31      file >> s >> wi >> s >> he;
32      cout << fixed << setprecision(2) << wi * he << el;
33  }
34  else
35  {
36      double side; file >> s >> side;
37      cout << fixed << setprecision(2) << side * side << el;
38  }
39  }
40  return 0;
41  }
42

```

	Input	Expected	Got	
✓	1.txt	28.09 80.00 13.51	28.09 80.00 13.51	✓
✓	2.txt	8.17 32.49 30.25 7.68 27.02 7.68	8.17 32.49 30.25 7.68 27.02 7.68	✓
✓	3.txt	8.17	8.17	✓
✓	4.txt	32.49	32.49	✓
✓	5.txt	7.68	7.68	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 3

Correct

Mark 10.00 out of 10.00

[RocketScience]

Bạn nhận được một tệp văn bản chứa tọa độ điểm xuất phát, tọa độ mục tiêu và vận tốc của một tên lửa (giả sử tên lửa bay thẳng với vận tốc không đổi). Hãy viết một hàm tính thời gian tên lửa cần để bay tới tiêu diệt mục tiêu của mình: `double estimatedTime(const char* path)`. Hàm nhận đầu vào là xâu kí tự chứa đường dẫn tới tệp thiết lập tên lửa và trả về số giây tên lửa hoàn thành hành trình. Đơn vị đo khoảng cách là *km* và đo tốc độ là *km/s*.

Tệp thiết lập mẫu có cấu trúc như sau:

```
SRC X: 0 Y: 0
DES X: 10000 Y: 10000
VELOCITY: 6 KM/S
```

For example:

Input	Result
1.txt	2357.02

Answer: (penalty regime: 0 %)

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 double estimatedTime(const char* path)
4 {
5     ifstream file(path);
6     string tmp;
7     double x1, x2, y1, y2, v;
8     file >> tmp >> tmp >> x1 >> tmp >> y1;
9     file >> tmp >> tmp >> x2 >> tmp >> y2;
10    file >> tmp >> v >> tmp;
11    return sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2)) / v;
12 }
```

	Input	Expected	Got	
✓	1.txt	2357.02	2357.02	✓
✓	2.txt	1566.94	1566.94	✓
✓	3.txt	0.00	0.00	✓
✓	4.txt	24961840.63	24961840.63	✓
✓	5.txt	2441396994689.99	2441396994689.99	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 4

Correct

Mark 10.00 out of 10.00

[Grade10]

Cho một bảng danh sách sinh viên và điểm của sinh viên đó như sau:

STT	Ho	Ten	Ngày Sinh	Dai so	THCS4	THCS1
1	Truong	Vo Ky	10/10/1999	8	10	9
2	Chu	Chi	Nhuoc 13/9/1999	10	10	10

Bảng đó được lưu trong tệp `.csv`:

```
STT,Ho,Ten,Ngày sinh,Dai so,THCS4,THCS1
1,Truong Vo,Ky,10/10/1999,8,10,9
2,Chu Chi,Nhuoc,13/9/1999,10,10,10
```

Hãy viết hàm `void grade10(const char* path)` nhận tham số đầu vào là một chuỗi kí tự chứa đường dẫn đến tệp `.csv` và in ra đầy đủ họ tên của các sinh viên có điểm THCS4 là 10 (in theo thứ tự trong bảng).

For example:

Input	Result
1.csv	Truong Vo Ky Chu Chi Nhuoc

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 void grade10(const char* path)
12 {
13     ifstream file(path);
14     string tmp;
15     getline(file, tmp);
16     /*while(file >> tmp)
17     {
18         for(int i = 0; i < int(tmp.size()); i++) if(tmp[i] == ',') tmp[i] = ' ';
19         stringstream ss(tmp);
20         string temp, ho, dem, ten, diem;
21         ss >> ho >> dem >> ten >> temp >> temp >> diem >> temp;
22         if(diem == "10") cout << ho << " " << dem << " " << ten << el;
23     }*/
24     while(getline(file, tmp))
25     {
26         string s, ho, ten, diem;
27         stringstream ss(tmp);
28         getline(ss, s, ',');
29         getline(ss, ho, ',');
30         getline(ss, ten, ',');
31         getline(ss, s, ',');
32         getline(ss, s, ',');
33         getline(ss, diem, '.');
```



```
34 |         getline(ss, s, ',');
35 |         if(diem == "10") cout << ho << " " << ten << endl;
36 |     }
37 | }
```

	Input	Expected	Got	
✓	1.csv	Truong Vo Ky Chu Chi Nhuoc	Truong Vo Ky Chu Chi Nhuoc	✓
✓	2.csv			✓
✓	3.csv	Ha Anh E Doan The A Kieu Van F Vu Hong T	Ha Anh E Doan The A Kieu Van F Vu Hong T	✓
✓	4.csv	Nguyen Van A Tran Minh C	Nguyen Van A Tran Minh C	✓
✓	5.csv			✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.



Question 5

Correct

Mark 10.00 out of 10.00

[MoveZeroes]

Viết hàm `void moveZeroes (int * arr, char * file)` thực hiện các công việc sau:

- Đọc các số nguyên dương từ tệp *file* vào mảng *arr*. Biết dòng đầu tiên của tệp lưu số nguyên *n* là số số nguyên dương, dòng tiếp theo chứa *n* số nguyên dương cần lưu vào mảng.
- Chuyển tất cả các phần tử của mảng có giá trị bằng 0 về cuối dãy, đồng thời, giữ nguyên vị trí tương đối của các phần tử còn lại trong mảng
- In dãy thu được ra màn hình

Chú ý: Bạn chỉ phải định nghĩa hàm theo yêu cầu mà **không** phải viết hàm `main()`, các thư viện cũng đã có sẵn.

For example:

Input	Result
first_array	272 881 478 93 645 14 0

Answer: (penalty regime: 0 %)

```

1  #include<bits/stdc++.h>
2  void moveZeroes (int * arr, char * file)
3  {
4      ifstream filepath(file);
5      int n; filepath >> n;
6      string tmp;
7      while(getline(filepath, tmp))
8      {
9          stringstream ss(tmp);
10         vector<string> vt1, vt2;
11         string s;
12         while(ss >> s)
13         {
14             if(s == "0") vt1.push_back(s);
15             else vt2.push_back(s);
16         }
17         for(auto x : vt2) cout << x << " ";
18         for(auto x : vt1) cout << x << " ";
19     }
20 }
```

	Input	Expected	Got	
✓	first_array	272 881 478 93 645 14 0	272 881 478 93 645 14 0	✓

	Input	Expected	Got	
✓	second_array	921 362 196 13 419 145 168 353 720 862 265 161 297 236 947 342 79 188 313 769 680 214 601 513 431 214 0 0 0 0 0 0 0 0	921 362 196 13 419 145 168 353 720 862 265 161 297 236 947 342 79 188 313 769 680 214 601 513 431 214 0 0 0 0 0 0 0 0	✓
✓	third_array	777 659 208 489 419 350 616 530 746 418 765 292 920 591 917 708 694 975 966 289 428 846 267 269 22 638 11 284 809 755 532 668 135 744 602 52 95 610 705 498 856 509 916 406 960 163 905 502 97 560 947 226 0 0 0 0 0 0 0 0 0 0 0 0	777 659 208 489 419 350 616 530 746 418 765 292 920 591 917 708 694 975 966 289 428 846 267 269 22 638 11 284 809 755 532 668 135 744 602 52 95 610 705 498 856 509 916 406 960 163 905 502 97 560 947 226 0 0 0 0 0 0 0 0 0 0 0 0	✓
✓	forth_array	247 269 521 747 228 257 207 123 107 653 71 89 418 33 327 851 570 551 807 153 279 847 484 886 93 227 693 790 79 647 990 535 447 999 223 897 175 393 560 977 458 96 922 765 820 120 350 439 341 878 24 156 169 222 874 587 384 968 563 410 571 363 677 53 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	247 269 521 747 228 257 207 123 107 653 71 89 418 33 327 851 570 551 807 153 279 847 484 886 93 227 693 790 79 647 990 535 447 999 223 897 175 393 560 977 458 96 922 765 820 120 350 439 341 878 24 156 169 222 874 587 384 968 563 410 571 363 677 53 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 6

Correct

Mark 10.00 out of 10.00

[MaxProductOfThree]

Với một dãy số nguyên cho trước, hãy viết hàm `int maximumProduct (char * path)` tìm bộ ba số có tích lớn nhất trong tất cả các bộ số và trả về tích lớn nhất đó.

Hàm nhận đầu vào là đường dẫn đến tệp lưu mảng số nguyên cho trước. Hàm trả về tích lớn nhất có thể có.

Dòng đầu tiên của tệp chứa kích cỡ n , dòng kế tiếp chứa n số nguyên cách nhau bởi một khoảng trắng.

Answer: (penalty regime: 0 %)

```
1 int maximumProduct (char * path)
2 {
3     ifstream file(path);
4     int n; file >> n;
5     int a[n];
6     for(auto &x : a) file >> x;
7     sort(a, a + n);
8     return max(a[0] * a[1] * a[n - 1], a[n - 1] * a[n - 2] * a[n - 3]);
9 }
```

	Test	Input	Expected	Got	
✓	maximumProduct(path)	1.txt	80	80	✓
✓	maximumProduct(path)	2.txt	448	448	✓
✓	maximumProduct(path)	3.txt	378	378	✓
✓	maximumProduct(path)	4.txt	103776	103776	✓
✓	maximumProduct(path)	5.txt	658240	658240	✓
✓	maximumProduct(path)	6.txt	922082	922082	✓

	Test	Input	Expected	Got	
✓	maximumProduct(path)	7.txt	1845738	1845738	✓
✓	maximumProduct(path)	8.txt	41897450	41897450	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 7

Correct

Mark 10.00 out of 10.00

[NeedForSpeed]

Frank Martin sau khi thôi không làm người vận chuyển nữa thì anh chuyển sang làm tài xế thử xe cho hãng xe. Công việc hàng ngày của anh là chạy những chiếc xe mới của hãng vài vòng và tính giờ. Kết quả được ghi trong một bảng biểu và được lưu trữ trong một tệp .csv. Cột đầu tiên của bảng là số thứ tự vòng đua (lap) và cột tiếp theo là thời gian Frank hoàn thành vòng đua đó, ví dụ một bảng và [cấu trúc](#) tệp tương ứng như sau:

Lap Time

1 143.23

2 134.12

Lap, Time

1,143.23

2,134.12

Cuối cùng, Frank cần phải tính thời gian trung bình anh hoàn thành vòng đua. Hãy giúp Frank bằng cách viết hàm **double** `averageTime(const char* path)`, nhận tham số là [xâu kí tự](#) lưu đường dẫn tới tệp .csv và trả về thời gian trung bình hoàn thành vòng đua. Lưu ý, tùy tâm trạng Frank sẽ chạy thử xe nhiều vòng hay ít vòng, nên số vòng đua trong các tệp .csv là không cố định.

For example:

Input	Result
1.csv	138.68

Answer: (penalty regime: 0 %)

```
1 #include<bits/stdc++.h>
2 double averageTime(const char* path)
3 {
4     ifstream file(path);
5     string s;
6     double ans = 0;
7     int dem = 0;
8     getline(file, s);
9     while(getline(file, s))
10    {
11        for(int i = 0; i < int(s.size()); i++) if(s[i] == ',') s[i] = ' ';
12        stringstream ss(s);
13        string x, y;
14        ss >> x >> y;
15        ans += stod(y);
16        dem ++;
17    }
18    return ans / dem;
19 }
```

	Input	Expected	Got	
✓	1.csv	138.68	138.68	✓
✓	2.csv	233.91	233.91	✓
✓	3.csv	333.22	333.22	✓
✓	4.csv	137.00	137.00	✓
✓	5.csv	83.15	83.15	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.



Question 8

Correct

Mark 10.00 out of 10.00

[NotBoringMovies]

Thành phố X mở một rạp chiếu phim mới, rất nhiều người muốn đến xem thử và hưởng các khuyến mại trong ngày khai trương.

Để đưa cho khách hàng các lựa chọn, rạp chiếu phim treo các poster về phim cùng với điểm đánh giá IMDB (rating) và mô tả của khách hàng về bộ phim đó.

Hãy viết hàm `void findMovies (char * path)` giúp khán giả tìm ra phim với điểm đánh giá lớn hơn 8.0 và có mô tả không "nhàm chán" ("**boring**").

Hàm nhận đầu vào là đường dẫn đến tệp chứa các poster phim. Hàm in ra tên những phim thỏa mãn điều kiện trên. Nếu không tìm thấy phim nào phù hợp, in ra thông báo "**All are boring!**".

Tệp có [cấu trúc](#) như sau:

```
id, movie, description, rating
1, Thor, great 3D, 8.0
2, The Rings, Fantasy, 7.5
3, Irish, boring, 6.2
4, Ice song, Fantasy, 8.6
5, Home Alone, Interesting, 6.5
```

For example:

Test	Input	Result
findMovies(path)	1.txt	Thor Ice song

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 void findMovies (char * path)
12 {
13     ifstream file(path);
14     string tmp;
15     getline(file, tmp);
16     vector<string> vt;
17     while(getline(file, tmp))
18     {
19         string id, mo, de, ra;
20         stringstream ss(tmp);
21         getline(ss, id, ',');
22         getline(ss, mo, ',');
23         getline(ss, de, ',');
24         getline(ss, ra, ',');
25         //cout << id << " " << mo << " " << de << " " << ra << el;
26         for(int i = 0; i < int(de.size()); i++)
27         {
28             if(isupper(de[i])) de[i] = islower(de[i]);
29         }
30         if(de != " boring" && stod(ra) >= 8.0) vt.push_back(mo);
```



```
31     }
32     if(int(vt.size()) != 0)
33     {
34         for(auto x : vt)
35         {
36             stringstream ss(x);
37             string y;
38             while(ss >> y) cout << y << " ";
39             cout << el;
40         }
41     }
42     else
43     {
44         cout << "All are boring!";
45     }
46 }
```

	Test	Input	Expected	Got	
✓	findMovies(path)	1.txt	Thor Ice song	Thor Ice song	✓
✓	findMovies(path)	2.txt	Kingsman	Kingsman	✓
✓	findMovies(path)	3.txt	Roman Holiday Pirates of the Caribbean: The Curse of the Black Pearl	Roman Holiday Pirates of the Caribbean: The Curse of the Black Pearl	✓
✓	findMovies(path)	4.txt	Hannah Montana The Movie Pirates of the Caribbean: The Curse of the Black Pearl	Hannah Montana The Movie Pirates of the Caribbean: The Curse of the Black Pearl	✓
✓	findMovies(path)	5.txt	All are boring!	All are boring!	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 9

Correct

Mark 10.00 out of 10.00

[ProductExceptSelf]

Với một dãy gồm n số nguyên với $n > 1$, giá trị của từng phần tử tại vị trí i được tính lại bằng tích của tất cả các phần tử còn lại trong dãy số nguyên trừ chính bản thân nó - *Product of Array Except Self*

Viết hàm `void productExceptSelf (char * path)` tính và in ra màn hình dãy số nguyên sau khi được tính lại theo qui tắc trên.

Hàm nhận đầu vào là đường dẫn *path* đến tệp chứa dãy số nguyên.

Chú ý: Bạn chỉ phải viết hàm mà không cần viết main(), các thư viện cũng đã được khai báo trước.

Tệp có [cấu trúc](#) như sau: Dòng đầu tiên chứa kích thước n . Dòng tiếp theo chứa n số nguyên được ngăn cách bởi dấu cách.

```
5
4 6 2 5 2
```

For example:

Test	Input	Result
productExceptSelf(path)	1.txt	-192 -240 320 240 240

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 void productExceptSelf (char * path)
12 {
13     ifstream file(path);
14     int n; file >> n;
15     ll a[n];
16     for(ll &x : a) file >> x;
17     if(n == 1) cout << a[0];
18     else
19     {
20         for(int i = 0; i < n; i++)
21         {
22             ll ans = 1;
23             for(int j = 0; j < n; j++)
24             {
25                 if(j != i) ans *= a[j];
26             }
27             cout << ans << " ";
28         }
29     }
30 }
```

	Test	Input	Expected	Got	
✓	productExceptSelf(path)	1.txt	-192 -240 320 240 240	-192 -240 320 240 240	✓
✓	productExceptSelf(path)	2.txt	-2352 2352 -1344 4704 9408 -1344 -1568	-2352 2352 -1344 4704 9408 -1344 -1568	✓
✓	productExceptSelf(path)	3.txt	-40 -20 8 -40 -10	-40 -20 8 -40 -10	✓
✓	productExceptSelf(path)	4.txt	61250 12250 -8750 -30625 12250 -12250 12250 -8750	61250 12250 -8750 -30625 12250 -12250 12250 -8750	✓
✓	productExceptSelf(path)	5.txt	81648 -63504 285768 190512 -571536 -63504 95256 -142884 81648	81648 -63504 285768 190512 -571536 -63504 95256 -142884 81648	✓
✓	productExceptSelf(path)	6.txt	-1	-1	✓
✓	productExceptSelf(path)	7.txt	0 0 0 0 0 0 -120960 0	0 0 0 0 0 0 -120960 0	✓
✓	productExceptSelf(path)	8.txt	24494400 27556200 -36741600 73483200 -31492800 -44089920 55112400 24494400 24494400 73483200 -44089920	24494400 27556200 -36741600 73483200 -31492800 -44089920 55112400 24494400 24494400 73483200 -44089920	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 10

Correct

Mark 10.00 out of 10.00

[HouseRobber]

Bạn là một kẻ trộm chuyên nghiệp và đang lên kế hoạch trộm những ngôi nhà dọc theo một con phố.

Mỗi ngôi nhà có một số tiền nhất định, sự ràng buộc duy nhất ngăn cản bạn lấy trộm là hai nhà liền kề có hệ thống an ninh kết nối và nó sẽ tự động liên lạc với cảnh sát nếu hai nhà liền kề đã bị đột nhập vào cùng một đêm.

Cho danh sách các số nguyên dương biểu diễn cho số tiền của mỗi ngôi nhà, xác định số tiền tối đa bạn có thể trộm tối nay mà không bị báo cảnh sát.

Viết hàm `int getMoney (int * houses, char * path)` để nhập số tiền chứa trong mỗi ngôi nhà từ tệp đầu vào được lưu theo đường dẫn `path` tương ứng với tên phố bạn định trộm. Số tiền của mỗi nhà được lưu cách nhau bởi một khoảng trắng và không biết trước số lượng nhà trong khu phố đó. Hàm trả về số lượng nhà sau khi bạn đã khởi tạo mảng `houses`.

Viết hàm `int rob(int * houses, int numOfHouses)` để tính và trả về số tiền lớn nhất mà bạn có thể trộm được từ con phố đó, biết số tiền trong mỗi ngôi nhà được lưu trong mảng `houses`

For example:

Input	Result
Montgomery_Street	23611

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 getMoney (int * houses, char * path)
13 {
14     ifstream file(path);
15     int x;
16     vector<int> vt;
17     while(file >> x)
18     {
19         vt.push_back(x);
20     }
21     int Size = vt.size();
22     for(int i = 0; i < Size; i++) houses[i] = vt[i];
23     return Size;
24 }
25
26 int rob(int * houses, int numOfHouses)
27 {
28     if(numOfHouses == 0) return 0;
29     if(numOfHouses == 1) return houses[0];
30     ll f[numOfHouses] = {0};
31     f[1] = max(houses[0], houses[1]);
32     f[0] = houses[0];
33     for(int i = 2; i < numOfHouses; i++)
34     {
35         f[i] = max (f[i - 2] + houses[i], f[i - 1]);
36     }
37     return f[numOfHouses - 1];
38 }
```

	Input	Expected	Got	
✓	Bailey_Avenue	816645	816645	✓
✓	Lawrence_Expressway	2610794	2610794	✓
✓	Montgomery_Street	23611	23611	✓
✓	San_Carlos	440	440	✓
✓	Mariposa	4444	4444	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Back to Course