

CSC10001

THI GIỮA KỲ (60')

April 2, 2024

Quy định thi giữa kỳ

- Được phép sử dụng tài liệu và Internet
- Đây là bài thi cá nhân, tất cả các hình thức gian lận đều được chuyển về Khoa với hình thức xử lý “RỐT MÔN”. Các hình thức gian lận gồm:
 - Trao đổi, thảo luận trong lúc làm bài thi
 - Sao chép bài người khác không phải của mình (của bạn, trên Internet, ...)
 - Cho bạn sao chép bài của mình
- Nộp tại Moodle môn học, phần Midterm
- Nộp file MSSV.zip/functions.cpp – chỉ nộp file FUNCTIONS.CPP

Hướng dẫn làm bài

- Thi giữa kỳ là thi trực tiếp trên phòng máy thực hành, thi bằng máy trường, tải đề bài và nộp bài trên Moodle
- Phải giải nén file đề thi và làm bài trên file đã giải nén
→ KHÔNG làm bài trên function.cpp vẫn còn nén trong file midterm.zip
- “Add” tất cả các file vào Empty C++ Project, và lập trình duy nhất trên **functions.cpp**, KHÔNG thay đổi các file khác
- Hoàn thành các hàm có **TODO** để có điểm, KHÔNG được thay đổi prototype của hàm đã có
- Sinh viên tự do thêm các lệnh/hàm con cần thiết để hoàn thành **TODO** list, tự do sử dụng những hàm được viết sẵn, tự do sử dụng thư viện/ thuật toán/ kiến thức đã học/đã biết
- **Lập vô tận = 0 điểm, lỗi không chạy được chương trình = 0 điểm (lỗi ở 1 hàm có thể làm cả chương trình không chạy được)**
→ comment/xóa tất cả các thân hàm gây lỗi trước khi nộp bài
→ KHÔNG xóa tên hàm vì sẽ gây lỗi chương trình
- Hàm **mainStudent** dùng để debug, sinh viên tùy ý sử dụng, thêm, xóa, sửa,... trong hàm này (hãy xem hàm này là **int main()** của bạn)
- Sinh viên sửa giá trị **volatile int flag = 1** để chạy bộ TEST và xem điểm của mình (giá trị khởi tạo là **flag = 0**).

- Đề thi gồm 3 câu, ứng với 3 hàm. Sinh viên nên đọc đề 1 lượt và chọn câu dễ hơn làm trước
- Thi trong 60' + 5' cuối giờ để nộp bài (tức là tổng cộng 65')
→ mỗi câu chỉ nên làm trong 20'
- Hết giờ nộp bài, cổng Moodle sẽ tự động khóa không cho nộp bài thêm nữa
→ **không nộp bài kịp giờ coi như không thi, không có ngoại lệ**
- Nếu làm trên linux và dịch bằng g++, câu lệnh để chạy :

```
>> g++ -o main main.cpp function.cpp  
>> ./main
```

Bài 1 (3)

Dãy số **Hailstone** bắt đầu từ một số tự nhiên n cho trước, các số tiếp theo sẽ được tạo theo quy tắc

- i) Nếu số hiện tại là chẵn thì số tiếp theo sẽ bằng số đó chia cho 2
- ii) Nếu số hiện tại là lẻ thì số tiếp theo sẽ được nhân lên 3 lần rồi cộng 1
- iii) Khi gặp số 1 thì dãy số kết thúc

Tính tổng các số trong dãy **Hailstone** khi biết n ?

- Input: $n = 3$
- Khi đó, dãy **Hailstone** tương ứng là 3, 10, 5, 16, 8, 4, 2, 1.
- Output: Tổng = 49
- Prototype: `int sumHailstone(int n)`
- Statement: `sumHailstone(3) = 49`

Bài 2 (3)

Cho bài toán cổ như sau: trâu đứng ăn năm, trâu nằm ăn ba, ba trâu già ăn một. Cho trước tổng số lượng trâu và cỏ, số lượng trâu đứng, trâu nằm, trâu già (khác 0) thỏa điều kiện lần lượt là (**standing**, **lying**, **oldbulls**). Hỏi có bao nhiêu bộ (**standing**, **lying**, **oldbulls**) thỏa điều kiện khi biết số lượng trâu và cỏ lần lượt là **buffaloes** và **bunches_of_grass**?

- Input: **buffaloes** = 100, **bunches_of_grass** = 120
 - `standing*5 + lying*3 + oldbulls/3 = bunches_of_grass`
 - `standing + lying + oldbulls = buffaloes`
 - `standing, lying, oldbulls > 0`
- Vét cạn (bruteforce) tất cả các khả năng có thể để chọn được các khả năng thỏa điều kiện
- Output: 5 bộ (**standing**, **lying**, **oldbulls**) thỏa là (2, 29, 69) (6, 22, 72) (10, 15, 75) (14, 8, 78) (18, 1, 81)
- Prototype: `int bruteforce(int buffaloes, int bunches_of_grass)`
- Statement: `bruteforce(100, 120) = 5`

Bài 3 (4)

Vé xem chung kết thể giới League of Legends là một số có chẵn chữ số (có 2, 4, 6, 8, ... chữ số). Nếu số trên vé của bạn là số may mắn, bạn sẽ nhận được một phần quà đặc biệt từ ban tổ chức. Số may mắn được định nghĩa là số có tổng các chữ số ở nửa đầu bằng tổng các chữ số ở nửa sau. Khán giả đến xem sẽ quẹt mã vé tại cửa vào.

Nằm trong đội IT tổ chức sự kiện, bạn hãy viết hàm kiểm tra xem vé vào cổng của khán giả có phải là vé may mắn hay không?

- Input: `ticket = 132222`
- Output: có phải là vé may mắn không
- Prototype: `int checkLuckyTicket(int ticket)`
- Statement: `checkLuckyTicket(132222) = 1`
- Hint: số kí tự của số `n` có thể được tính bằng:
`int digits = (int)log10(n) + 1;`
VD: với `n = 123`, $\rightarrow \log_{10}(123) = 2.0899$,
 \rightarrow lấy phần nguyên của nó $(\text{int})2.0899 = 2$, và cộng 1 để được số kí tự của `n`