

Tên học phần: Kỹ thuật Lập trình Mã HP: CSC10002  
Thời gian làm bài: 75 phút Ngày thi: 25/10/2021  
Ghi chú: Sinh viên [ ☒ được phép / ☐ không được phép ] sử dụng tài liệu khi làm bài.

Họ tên sinh viên: ..... MSSV: ..... STT: .....

Gọi A, B, C, D là 4 chữ số cuối trong mã SV của bạn; đặt  $M=10+A$ ,  $N=21-B$ ,  $P=A+B$ ,  $Q=M+N$   
( Ví dụ, nếu mã SV là 20123456 thì  $A=3$ ,  $B=4$ ,  $C=5$ ,  $D=6$ ; và  $M=13$ ,  $N=17$ ,  $P=7$ ,  $Q=30$  )

**Câu 1\_AC** (0.5 điểm - dành cho các SV có A là số chẵn).

Tìm một bộ giá trị cho X và Y để đoạn code sau xuất ra giá trị Q, giải thích lý do

```
int *p1 = &X, *p2 = &Y;  
*p1 -= M; *p2 += N;  
cout<< P + *p1 + *p2;
```

**Câu 1\_AL** (0.5 điểm - dành cho các SV có A là số lẻ).

Hãy cho biết đoạn code sau xuất ra giá trị gì, giải thích lý do

```
int X = M, Y = N, *p1 = &X, *p2 = &Y;  
*p1 -= P; *p2 += P;  
cout<< *p1 + *p2;
```

**Câu 2\_BC** (0.5 điểm - dành cho các SV có B chẵn).

Hãy viết hàm F để dòng code bên dưới có kết quả  $X=C$  và  $Y=P$

```
int X = A, Y = X + F(X);
```

**Câu 2\_BL** (0.5 điểm - dành cho các SV có B lẻ).

Hãy viết hàm F để dòng code bên dưới có kết quả  $X=D$  và  $Y=Q$

```
int X = B, Y = X - F(X);
```

**Câu 3\_CC** (1 điểm - dành cho các SV có C chẵn).

Hãy viết hàm F để đoạn code bên dưới có kết quả  $Y=M$

```
int X = A, Y = B, *p1 = &X, *p2 = &Y;  
F(p1, p2);  
Y = X + *p1 - F(p2, p1);
```

**Câu 3\_CL** (1 điểm - dành cho các SV có C lẻ).

Hãy viết hàm F để đoạn code bên dưới có kết quả  $Y=N$

```
int X = A, Y = B, *p1 = &X, *p2 = &Y;  
F(p1, p2);  
Y = X - *p2 + F(p1, p2);
```

**Câu 4\_DC** (1 điểm - dành cho các SV có D chẵn).

Hãy giải thích ý nghĩa của từng dòng code bên dưới, cho biết có thể xảy ra vấn đề bất ổn gì khi hàm chạy và cách khắc phục tương ứng

```
bool CreateArray (Array &A, int num) {
    A.n = num;
    A.capacity = A.n + RESERVE;
    A.items = new int [A.capacity];
    return (A.items != NULL);
}
```

**Câu 4\_DL** (1 điểm - dành cho các SV có D lẻ).

Hãy giải thích ý nghĩa của từng dòng code bên dưới, cho biết có thể xảy ra vấn đề bất ổn gì khi hàm chạy và cách khắc phục tương ứng

```
void inputArr (Array &A) {
    A.n = 0;
    do {
        cout<< " Phan tu ["<<A.n<<"] = "; cin>> A.items[n];
        A.n++;
        if (A.n==A.capacity ) return;
        cout<< " Co nhap tiep khong ? (C/K) ";
    }while (getchar() == 'C');
}
```

**Câu 5\_MC** (1 điểm - dành cho các SV có B chẵn).

Thao tác thêm 01 phần tử vào vị trí A trên mảng có nhanh hơn so với trên xâu đơn? Lý do?

**Câu 5\_ML** (1 điểm - dành cho các SV có B lẻ).

Thao tác xóa 01 phần tử ở vị trí B trên mảng có nhanh hơn so với trên xâu đơn? Lý do?

**Câu 6\_NC** (1 điểm - dành cho các SV có N chẵn).

Xâu đơn luôn có con trỏ lưu vị trí phần tử cuối có phù hợp với cấu trúc ngăn xếp không? Lý do?

**Câu 6\_NL** (1 điểm - dành cho các SV có N lẻ).

Xâu đơn luôn có con trỏ lưu vị trí phần tử cuối có những hạn chế gì so với xâu không có con trỏ này?

**Câu 7\_PC** (1 điểm - dành cho các SV có P chẵn).

Tập tin văn bản có ưu điểm gì so với tập tin nhị phân? Viết hàm lưu thêm 01 số nguyên vào một tập tin văn bản và hàm đọc 01 số nguyên ở đầu một tập tin nhị phân.

**Câu 7\_PL** (1 điểm - dành cho các SV có P lẻ).

Tập tin văn bản có khuyết điểm gì so với tập tin nhị phân? Viết hàm lưu thêm 01 số nguyên vào một tập tin nhị phân và hàm đọc 01 số nguyên ở đầu một tập tin văn bản.

**Câu 8** (1 điểm). Cho biết 2 hàm bên dưới có công dụng gì, chỉ ra các khác biệt khi 2 hàm được chạy

<pre>void F1 (Node * &amp;L) {     Node * L2 = NULL, *p = L;     while ( L != NULL ) {         L = L-&gt;Next;         p-&gt;Next = L2;         L2 = p;         p = L;     }     L = L2; }</pre>	<pre>void F2 (Node * &amp;L) {     Node * L2 = 0;     while (L) {         Node *p = L;         L = L-&gt;Next;         p-&gt;Next = L2;         L2 = p;     }     L = L2; }</pre>
--	---

**Câu 9 (2 điểm).**

Một đứa trẻ được theo dõi sự tăng trưởng bằng cách cứ đến một ngày nào đó lại được cân đo và lưu lại 3 thông tin: thời điểm (*bao gồm ngày – tháng – năm*), chiều cao (*theo mét, chính xác đến 0.01*), cân nặng (*theo kg, chính xác đến 0.1*).

Giả sử đã có sẵn một loạt thông tin cân đo của một bé (*đã được ghi trong 1 sổ sức khỏe*). Và người ta có nhu cầu làm một chương trình thực hiện các xử lý:

- Nhập vào chương trình tất cả các thông tin trên để có một danh sách được lưu trữ theo đúng thứ tự thời gian (*dù người nhập có thể không nhập đúng trật tự*).
- Xuất lại các thông tin theo thứ tự ngược dòng thời gian (*thời điểm gần hiện tại hơn thì xuất trước*), trong đó các tháng có nhiều thông tin thì chỉ xuất 1 thông tin.
- Xác định (và xuất ra) những tháng không có thông tin.
- Xác định những tháng có nhiều thông tin và loại bỏ tất cả - chỉ giữ lại một (*giữ lại phần tử có ngày nhỏ nhất trong tháng đó*).
- Xác định những thời điểm bất hợp lý (*không theo đúng thứ tự thời gian hoặc ngày-tháng-năm không hợp lệ*) và loại bỏ phần tử tương ứng ra khỏi danh sách.
- Xác định 2 mốc thời gian mà số ngày bé không được cân đo là dài nhất
- Xác định những thời điểm mà chỉ số BMI ra ngoài ngưỡng [A..B] (*chỉ số BMI được tính bằng  $\frac{\text{cân\_nặng}}{(\text{chiều\_cao} * \text{chiều\_cao})}$* ).
- Xác định giai đoạn (*khoảng thời gian giữa 2 thời điểm kế nhau*) mà chiều cao hoặc cân nặng tăng trưởng nhanh nhất /chậm nhất.
- Nhập thêm 1 thông tin cân đo và thông tin vừa nhập được đưa vào đúng vị trí trong danh sách đã có.

a/ Hình thức biểu diễn nào là hợp lý nhất để lưu trữ và xử lý các số liệu trên trong chương trình C (*khai báo cụ thể*)? Lý do?

b/ Viết 01 hàm C/C++ thực hiện một trong các xử lý trên theo Kiểu dữ liệu mà bạn đã chọn ở câu (a).

Ghi chú : Một ví dụ về các thông tin cân đo của một em bé :

Thời điểm	Cân nặng (kg)	Chiều cao (m)
28/7/2009	3.3	0.51
15/8/2009	4.0	0.54
27/8/2009	4.7	0.57
1/1/2010	8.0	0.68
1/6/2010	10.0	0.77

(với thông tin như bảng trên thì:

+ 2 mốc thời gian mà số ngày bé không được cân đo dài nhất là 1/1/2010 và 1/6/2010;

+ tháng 8/2009 có 2 thông tin và khi lọc thì sẽ bỏ phần tử ứng với ngày 29/8/2009)

----- Hết. -----