Thời gian**:** 90 phút

**Câu 1**. Những mệnh đề nào dưới đây là đúng: (1,5 điểm)

• Ngôn ngữ C++ có thể sử dụng các hàm của C như memcpy, strlen, …

• Cùng một tên biến có thể khai báo trong cácđoạn khác nhau của một khối {…}

• Từ long không thể dùng làm tên biến

• Toán tử & cho giá trị 0 hoặc 1

• Nếu khai báo int \*p = (int \*)1090 thì cout << (p + 1) đưa ra màn hình giá trị 0x0000044A

• Toán tử return ((char \*)p) trả giá trị p về cho hàm kiểu char \*

• Tham số mặc định của hàm phải đặt ở cuối danh sách tham số

• Định nghĩa struct So{double a}; không có ý nghĩa thực tiễn

• Độ phức tạp trung bình của thuật toán sắp xếp nổi bọt (BubbleSort) là O(n2)

• Một hàm đệ quy không bao giờ dẫn đến lỗi tràn stack

**Câu 2**. Cho các biến sau: (2 điểm)

char c; int i; unsigned u; double d; int &ri = i; double \*pd = &d;

Cho biết giá trị của các biến bị thay đổi sau khi thực hiện lần lượt từng dòng lệnh

c = 'A'; i = 5; u = 1; d = 1.5; ri = 2; \*pd = 5;

c = 'A'; i = 2; u = 1; d = 5; ri = 2; \*pd = 5;

if (++u || ++i || ++c) d /= 4;

u = 2; i = 2; c ='A'; d = 1.25

for (; i > 0; i--) {u += (i << 1); u--;}

i = 2; u = 2 + (2<<1) = 2 + 4 = 6; u = 5

i = 1; u = 5 + (1<<1) = 5 + 2 = 7; u = 6;

i = 0;

switch(c){case 'A':d++;case 'B':d+=2;default:d+=5;}

c ='A'; d = 2.25; d = 4.25; d = 9.25

Kết quả cuối cùng: c = 'A'; i = 0; u = 6; d = 9.25; ri = 0; \*pd = 9.25;

**Câu 3**. Cho đoạn khai báo sau: (1,5 điểm)

char s[]="1234567890"; unsigned \*p = (unsigned \*)&s[2];

a) Mảng s có 10 phần tử:  đúng  sai //có 11 phần tử

b) (char)(\*(p + 1)) bằng '5':  đúng  sai //bằng 7

c) Nếu gán \*p=0x4163 thì cout << s cho hiện lên màn hình: 12cA

Câu 4. Hàm nào dưới đây tính x^n: (1 điểm)

double f1(double x, int n)

{

double res = 1.0;

while (n) {res \*= x; --n;}

return (res);

}

double f2(double x, int n)

{

if (n < 1)

return 1.0;

return (x \* f2(x, n – 1));

}

 Hàm f1

 Hàm f2

 Cả hai hàm

**Câu 5**. Điền biểu thức vào những chỗ còn thiếu trong các hàm dưới đây (2 điểm)

// Hàm tìm độ dài của xâu ký tự

int len(char \*s) {int k = 0; while (s[k]!= '\0') k++; return (k);}

// Hàm so sánh hai xâu ký tự

int cmp(char \*s1, char \*s2) {

for (int i = 0; i <= (len(s1) < len(s2) ? len(s1):len(s2) ); i++) {

if (s1[i] < s2[i]) return (-1);

if(s1[i] > s1[i]) return (1);

}

return 0; // hai xâu bằng nhau

}

// Hàm nối hai xâu ký tự

char\* concat(char\* s1, char\* s2) {

int l1 = len(s1), l2 = len(s2);

char\* s = new char[l1 + l2 +1], \*p = s, \*q;

for (q = s1; \*q; q++, p++) \*p = \*q;

for (q = s2; \*q; q++,p++ ) \*p = \*q;

\*p = 0;

return (s);

}

**Câu 6**. Cho đoạn chương trình dưới đây (2 điểm)

struct date {

int day, month, year;

int compare(int d, int m, int y) {

int d1 = (year << 9) + (month << 5) + day;

int d2 = (y << 9) + (m << 5) + d;

if (d1 < d2)

return -1;

return (d1 > d2);

}

void set(int d, int m, int y) { day = d; month = m; year = y; }

};

struct student {

int id; char name[50]; date dob; // mã, họ tên và ngày sinh

student \*prev, \*next;

void Create(int i, char \*n, int d, int m, int y) {

next = prev = NULL;

id = i;

memcpy(name, n, len(n)+1);

dob.set(d, m, y);

}

void Print() { cout << id << '\t' << name << '\t' <<

dob.day << '/' << dob.month << '/' << dob.year << endl; }

};

struct list {

student \*head, \*tail;

void Init() { head = tail = NULL; }

void Insert(int id, char \*name, int d, int m, int y);

void Print();

};

a) Hoàn thành hàm Insert của struct list để thêm một sinh viên vào đầu danh sách

void Insert(int id, char \*name, int d, int m, int y){

student \*A = (student \*)malloc(sizeof(student));

A->Create(id, name, d, m, y);

if(head == NULL){

head = tail = A;

}

else{

A->next = head;

head = A;

}

}

b) Hoàn thành hàm Print của struct list để in ra danh sách những sinh viên sinh sau ngày 01/01/1980

void Print(){

for(student \*p = head; p != NULL; p = p->next ){

if(p->date.compare(1,1,1980)==1){

p->Print();

}

}

}