

Tổng kết môn học

Các nội dung chính

- (1) Tổng quan về kỹ thuật lập trình,
- (2) Quản lý bộ nhớ,
- (3) Hàm,
- (4) Kỹ thuật viết mã nguồn hiệu quả,
- (5) Phong cách lập trình,
- (6) Kỹ thuật đệ quy và khử đệ quy,
- (7) Kỹ thuật sử dụng cấu trúc dữ liệu cơ bản,
- (8) Kỹ thuật lập trình phòng ngừa,
- (9) Kỹ thuật gỡ rối, kiểm thử và tinh chỉnh mã nguồn.

Đánh giá học phần



Điểm thành phần	Phương pháp đánh giá cụ thể	Mô tả	CĐR được đánh giá	Tỷ trọng
[1]	[2]	[3]	[4]	[5]
A1. Điểm quá trình (*)	Đánh giá quá trình			40%
	A1.1. Bài tập về nhà	Tự luận	M2.1 M2.2	10%
	A1.2a. Bài tập nhóm	Báo cáo	M2.3 M1.4	30%
	A1.2b. Thi giữa kỳ	Tự luận và/ hoặc trắc nghiệm	M2.1 M2.2 M2.3	30%
A2. Điểm cuối kỳ	A2.1. Thi cuối kỳ	Tự luận và/ hoặc trắc nghiệm	M1.2 M1.4 M2.2 M2.3	60%

Nội dung chi tiết



STT	Lý thuyết
1	<ul style="list-style-type: none">• Chương 1: Tổng quan về kỹ thuật lập trình• 1.1. Giới thiệu về học phần kỹ thuật lập trình và các khái niệm cơ bản• 1.2. Mô thức lập trình• 1.3. Giới thiệu về ngôn ngữ C++• 1.4. Vào ra và biên dịch trong C++
2	<ul style="list-style-type: none">• Chương 2: Quản lý bộ nhớ• 2.1 Con trỏ và tham chiếu• 2.2 Mảng và con trỏ• 2.3 Mô hình tổ chức và quản lý bộ nhớ• 2.4 Cấp phát động
3	<ul style="list-style-type: none">• Chương 3. Hàm• 3.1 Truyền tham trị, tham biến và tham số ngầm định• 3.2 Đa năng hóa hàm• 3.3 Tham số hóa hàm• 3.4 Khái quát hóa hàm• 3.5 Biểu thức lamda và hàm ẩn danh

Nội dung chi tiết



STT	Lý thuyết
4	<ul style="list-style-type: none">• Chương 4: Kỹ thuật viết mã nguồn hiệu quả• 4.1 Các kỹ thuật viết mã nguồn hiệu quả• 4.2 Những nguyên tắc cơ bản trong việc tăng hiệu quả viết mã nguồn• 4.3 Tối ưu hóa mã nguồn C/C++
5	<ul style="list-style-type: none">• Chương 5. Phong cách lập trình• 5.1. Các quy tắc cơ bản• 5.2. Phong cách lập trình chuẩn mực• 5.3. Tổ chức và xây dựng chương trình – mô-đun hóa• 5.4. Viết tài liệu chương trình
6	<ul style="list-style-type: none">• Chương 6: Kỹ thuật đệ quy• 6.1 Nhắc lại khái niệm đệ quy và thuật toán đệ quy, các loại đệ quy và một số ví dụ minh họa• 6.2 Khử đệ quy

Nội dung chi tiết



STT	Lý thuyết
7	<ul style="list-style-type: none">• Chương 7. Ứng dụng các cấu trúc dữ liệu cơ bản• 7.1 Nhắc lại các cấu trúc dữ liệu cơ bản: danh sách liên kết, ngăn xếp, hàng đợi, cấu trúc liên kết (map, set)• 7.2 Giới thiệu thư viện STL và cách sử dụng các cấu trúc cơ bản: pair, vector, list, stack, queue, set, map...• 7.3 Các ví dụ ứng dụng cấu trúc dữ liệu cơ bản để giải quyết các bài toán cụ thể
8	<ul style="list-style-type: none">• Chương 8: Bẫy lỗi và lập trình phòng ngừa• 8.1 Khái niệm• 8.2 Bảo vệ chương trình khi dữ liệu đầu vào không hợp lệ (Invalid Inputs)• 8.3 Assertions• 8.4 Kỹ thuật xử lý lỗi• 8.5 Xử lý ngoại lệ
9	<ul style="list-style-type: none">• Chương 9: Kỹ thuật gỡ rối, kiểm thử và tinh chỉnh mã nguồn• 9.1 Kỹ thuật gỡ rối• 9.2 Kiểm thử đơn vị• 9.3 Tinh chỉnh mã nguồn

Bài tập

1

Trắc nghiệm

2

Tự luận

Trắc nghiệm

1

Khai báo biến nào sau đây **đúng**?

A

int length; float width;

B

int length, float width;

C

int length; width;

D

int length, int width;

Trắc nghiệm

2

Chọn **kết quả in ra màn hình** của đoạn chương trình sau:

A

x = 1, y = 2

B

x = 2, y = 1

C

x = 2, y = 2

D

Chương trình bị lỗi do x được khai báo lại

```
void main()
{
    int x = 1;
    int y = x = x + 1;
    printf("x = %d, y = %d", x, y);
}
```

Trắc nghiệm

3

Giả sử **x** là biến ký tự có giá trị 'b'.
Câu lệnh **printf("%c", ++x);** in ra:

A

a

B

b

C

c

D

d

Trắc nghiệm

4

Phân tích
đoạn mã sau:

```
#include <stdio.h>
void main()
{
    int i, j;
    printf("Enter an integer: ");
    scanf("%d", &j);
    i = i + 4;
}
```

A

Báo lỗi biên dịch vì i...

B

Báo lỗi biên dịch (error) vì j không được khởi tạo

C

Báo lỗi thực thi do i không có giá trị khởi tạo tại $i = i + 4$;

D

Chương trình biên dịch và thực thi bình thường

Trắc nghiệm

5

Giá trị cuối cùng của x bằng bao nhiêu khi x có giá trị khởi tạo là 1.

A

8

B

6

C

1

D

7

```
if (x >= 0)
    x += 5;
else if (x >= 5)
    x += 2;
```

Trắc nghiệm

6

Giả sử $x = 1$, $y = -1$, $z = 1$.

Cho biết **kết quả in ra** của đoạn chương trình sau:

A

$x > 0$ and $y > 0$

B

$x < 0$ and $z > 0$

C

$x < 0$ and $z < 0$

D

Không in gì cả

```
if (x > 0)
    if (y > 0)
        printf("x > 0 and y > 0");
else if (z > 0)
    printf("x < 0 and z > 0");
```

Trắc nghiệm

7

Cho biết **giá trị cuối cùng của x**.

A

1

B

2

C

3

D

4

```
int x = 3;  
if (x == 2);  
    x = 0;  
if (x == 3) x++;  
else x += 2;
```

Trắc nghiệm

8

y bằng bao nhiêu sau khi thực thi phát biểu switch sau:

A

1

B

2

C

3

D

4

```
int x = 3, y;  
switch (++x + 3) {  
    case 6: y = 0;  
    case 7: y = 1;  
    default: y += 1;  
}
```

Trắc nghiệm

9

Kết quả xuất ra của phát biểu switch dưới đây là gì?

A

abcd

B

bcd

C

bb

D

bbb

```
char ch = 'b';  
switch (ch) {  
    case 'a': printf("%c", ch);  
    case 'b': printf("%c", ch);  
    case 'c': printf("%c", ch);  
    case 'd': printf("%c", ch);  
}
```


Trắc nghiệm

10

Kết xuất cuối cùng của giá trị **S**?

A

12

B

10

C

32

D

Lặp vô hạn

```
int S = 0, i = 1;
while (i = 1) {
    S = S + 2 * i;
    i++;
    if (i >= 5 || S > 30)
        break;
}
printf("%d", S);
```

Trắc nghiệm

11

Kết xuất cuối cùng của giá trị **S**?

A

45

B

55

C

50

D

Lặp vô hạn

```
int i = 1, S = 0;
while (1) {
    S += i++;
    if (i % 10 == 0)
        break;
}
printf("%d", S);
```

Trắc nghiệm

12

sum bằng bao nhiêu sau khi vòng lặp dưới đây kết thúc?

A

5

B

6

C

7

D

8

```
int sum = 0;
int item = 0;
do {
    item++;
    sum += item;
    if (sum > 4)
        break;
} while (item < 5);
```

Trắc nghiệm

13

Cho biết đoạn chương trình sau đây **xuất ra màn hình những gì?**

A

1 2

B

1 3

C

0 2

D

0 1

```
int i, a = 0;
for (i = 0; i < 3; i++) {
    if (i == 2)
        continue;
    a += i;
    if (i > 1) break;
    printf("%d ", a);
}
```

Trắc nghiệm

14

Kết quả cuối cùng của giá trị **S** sau vòng lặp?

A

60

B

14

C

5

D

32

```
int i, S = 0;
for (i = 5; i < 20; i += 2) {
    S += i;
    if ((i%5 == 0) && (i%3 == 0))
        break;
}
printf("%d", S);
```

Trắc nghiệm

15

Kết quả cuối cùng của giá trị **S** sau vòng lặp

A

32

B

8

C

5

D

Kết quả khác

```
int S = 0, i;  
for (i = 5; i < 20; i += 2) {  
    switch (i % 2) {  
        case 0: S += S; break;  
        case 1: S += 1; break;  
    }  
}  
printf("%d", S);
```

Trắc nghiệm

16

Kết quả của đoạn chương trình sau là gì?

A

5 10

B

6 12

C

4 5

D

Kết quả khác

```
int a[] = {1, 2, 3, 4, 5}, N = 5, i;  
for (i = 0; i < N/2; i++)  
    a[i] = a[N - i + 1];  
printf("%d %d", a[3], a[4]);
```

Trắc nghiệm

17

Kết quả của đoạn chương trình sau?

A

3 4 1 5 2

B

1 2 3 4 5

C

5 4 3 2 1

D

5 3 1 2 3

```
int a[] = {3, 4, 1, 5, 2}, N=5, x, y, t;  
for (x = 0; x < N; x++)  
    for (y = 0; y < N; y++)  
        if (a[x] > a[y]) {  
            t = a[x];  
            a[x] = a[y];  
            a[y] = t;  
        }  
for (x = 0; x < N; x++)  
    printf("%d ", a[x]);
```


Trắc nghiệm

18

Chức năng của đoạn chương trình sau là gì?

A

Kô thay đổi a

B

Sắp xếp a giảm

C

Đảo ngược a

D

Tìm max của a

```
int a[] = {3, 4, 1, 5, 2}, N = 5, i, k;  
for (i = 0; i < N/2; i++) {  
    k = a[i];  
    a[i] = a[N - i - 1];  
    a[N - i - 1] = k;  
}  
for (i = 0; i < N; i++)  
    printf("%d ", a[i]);
```

Trắc nghiệm (tự trả lời)

19

Giả sử a và b là hai số thực. Biểu thức nào dưới đây viết không đúng theo cú pháp của ngôn ngữ lập trình C?

A

$(a -= b);$

B

$(a \&= b);$

C

$(a *= b);$

D

$(a += b);$

Trắc nghiệm (tự trả lời)

20

Giả sử a và b là hai số thực. Biểu thức nào dưới đây viết không đúng theo cú pháp của ngôn ngữ lập trình C?

A

$(a -= b);$

B

$(a += b);$

C

$(a >> = b);$

D

$(a /= b);$

Trắc nghiệm (tự trả lời)

21

Cho $a = 3$, $b = 2$. Biến $c = (a \ll b)$ sẽ có giá trị nào trong những giá trị dưới đây:

A

$c = 8$

B

$c = 9$

C

$c = 6$

D

$c = 12$

Trắc nghiệm (tự trả lời)

22

Cho $a = 3$, $b = 2$. Biến $c = (a > b)$ sẽ có giá trị nào trong những giá trị dưới đây:

A

$c = 1.5$

B

$c = 1$

C

$c = 0$

D

$c = -1$

Trắc nghiệm (tự trả lời)

23

Xâu định dạng nào dưới đây dùng để in ra một kí tự?

A

`"%s"`

B

`"%f"`

C

`"%x"`

D

`"%c"`

Trắc nghiệm (tự trả lời)

24

Xâu định dạng nào dưới đây dùng để in ra một số nguyên dài?

A

`"%ld"`

B

`"%ox"`

C

`"%od"`

D

`"%oo"`

Trắc nghiệm (tự trả lời)

25

Hãy cho biết giá trị của a và b sau khi thực hiện đoạn chương trình dưới đây

A

a = 17, b = 0;

B

a = 5, b = 1;

C

a = 3, b = 2;

D

a = 2, b = 3;

```
#include <stdio.h>
#include <conio.h>
int a = 5, b = 12;
void swap ( int a, int b) {
    a = a + b; b = a - b; a = a - b;
}
void main(void){
    int a = 3, b = 2;
    swap(a, b);
    printf("\n a= %d b= %d",a,b);
}
```


Trắc nghiệm (tự trả lời)

26

Hãy cho biết giá trị của a và b sau khi thực hiện đoạn chương trình dưới đây

A

a = 12, b = 5;

B

a = 12, b = 12;

C

a = 5, b = 12;

D

a = 5, b = 5;

```
#include <stdio.h>
#include <conio.h>
int a = 5, b = 12;
void swap ( int a, int b) {
    int temp = a; a = b; b = temp;
}
void main(void){
    swap(a, b);
    printf("\n a= %d b= %d",a,b);
}
```

Trắc nghiệm (tự trả lời)

27

Hãy cho biết kết quả thực hiện đoạn chương trình dưới đây

A

Ket qua: 13

B

Ket qua: 720

C

Ket qua: 7

D

Ket qua: 8

```
#include <stdio.h>
#include <conio.h>
int function1(int n){
    int f1=1, f2=1, fn,i=2;
    while(i<=n){
        fn=f1+f2; f1=f2;f2=fn;i++;}
    return(fn);
}
void main(void){
    printf("\nKet qua:%d",function1(7));
}
```

Trắc nghiệm (tự trả lời)

28

Hãy cho biết kết quả thực hiện đoạn chương trình dưới đây

A

Ket qua: 13

B

Ket qua: 21

C

Ket qua: 8

D

Ket qua: 24

```
#include <stdio.h>
#include <conio.h>
int function1(int n){
    if(n==1) return(1);
    else if(n==2) return(1);
    return(function1(n-1)+function1(n-2));
}
void main(void){
    printf("\nKet qua:%d",function1(8));
}
```

Trắc nghiệm (tự trả lời)

29

Dấu hiệu nào dưới đây cho biết node p của một danh sách liên kết đơn chưa phải là node cuối cùng bên phải:

A

(p -> next !=NULL)

B

(p -> infor !=NULL)

C

(p -> infor ==NULL)

D

(p -> next ==NULL)

Trắc nghiệm (tự trả lời)

30

Dấu hiệu nào dưới đây cho biết node p của một danh sách liên kết đơn là node cuối cùng bên phải:

A

(p ->infor ==NULL)

B

(p ->infor !=NULL)

C

(p -> next !=NULL)

D

(p -> next ==NULL)

Trắc nghiệm (tự trả lời)

31

Dấu hiệu nào dưới đây cho biết node p của một danh sách liên kết kép là node cuối cùng bên trái:

A

(p ->infor ==NULL)

B

(p ->infor !=NULL)

C

(p -> left !=NULL)

D

(p -> left ==NULL)

Trắc nghiệm (tự trả lời)

32

Dấu hiệu nào dưới đây cho biết node p của một danh sách liên kết kép chưa phải là node cuối cùng bên phải:

A

(p ->right ==NULL)

B

(p ->left !=NULL)

C

(p -> right !=NULL)

D

(p -> left ==NULL)

Trắc nghiệm (tự trả lời)

33

Từ một đỉnh bất kỳ của danh sách liên kết kép:

A

Ta duyệt được cả các node bên phải và bên trái danh sách

B

Ta chỉ duyệt được các node bên phải danh sách

C

Ta chỉ duyệt được các node bên trái danh sách

D

Ta không duyệt được các node bên phải và bên trái danh sách

Trắc nghiệm (tự trả lời)

34

Khi loại bỏ node p khỏi danh sách liên kết đơn thì:

A

Ta phải dịch chuyển về node gốc để thực hiện

B

Ta phải dịch chuyển đến node cuối cùng để thực hiện

C

Ta phải dịch chuyển về node sau node p để thực hiện

D

Ta phải dịch chuyển về node trước node p để thực hiện

Trắc nghiệm (tự trả lời)

35

Khi thêm một node p vào đầu danh sách liên kết đơn thì:

A

Ta chỉ cần thiết lập liên kết trái cho p

B

Ta lấy p làm gốc mới và thiết lập liên kết phải cho p

C

Ta thiết lập cả liên kết phải và liên kết trái cho p

D

Ta chỉ cần thiết lập liên kết phải cho p

Trắc nghiệm (tự trả lời)

36

Hãy cho biết kết quả thực hiện đoạn chương trình dưới đây

A

5

B

6

C

7

D

8

```
int a=3,b=0;  
b=a++;  
b=b+++a;  
cout<<b;
```

Trắc nghiệm (tự trả lời)

37

Hãy cho biết kết quả thực hiện đoạn chương trình dưới đây

A

2

B

3

C

4

D

5

```
int a=3,b=0;  
b=--a;  
b=b+--a;  
cout<<b--;
```

Trắc nghiệm (tự trả lời)

38

Hàm nào dưới đây chỉ dùng để đọc một khối dữ liệu trong file:

A

Hàm fgets()

B

Hàm fwrite()

C

Hàm fread()

D

Hàm fputs()

Trắc nghiệm (tự trả lời)

39

Hàm nào dưới đây chỉ dùng để đọc một kí tự từ tệp:

A

Hàm scanf()

B

Hàm fscanf()

C

Hàm fgetc()

D

Hàm getch()

Trắc nghiệm (tự trả lời)

40

Hàm nào dưới đây chỉ dùng để ghi một khối dữ liệu trong file:

A

Hàm fgets()

B

Hàm fwrite()

C

Hàm fread()

D

Hàm fputs()

Trắc nghiệm (tự trả lời)

41

Hàm nào dưới đây chỉ dùng để ghi một kí tự vào tệp:

A

Hàm putch()

B

Hàm puts()

C

Hàm fputs()

D

Hàm fputc()

Trắc nghiệm (tự trả lời)

42

Thực hiện hàm đệ quy không thoả mãn điều kiện dừng thì điều gì sẽ xảy ra:

A

Trình biên dịch báo lỗi

B

Không thể chạy được chương trình

C

Sẽ gây đụng độ với chương trình khác

D

Chương trình bị ngắt do tràn bộ nhớ

Trắc nghiệm (tự trả lời)

43

Phép cộng thêm 1 vào con trỏ đang trỏ đến một mảng sẽ:

A

Gây lỗi

B

Làm con trỏ trỏ đến phần tử tiếp theo của mảng

C

Tăng giá trị của phần tử mà con trỏ trỏ đến

D

Các phương án trên đều sai

Trắc nghiệm (tự trả lời)

44

Trong các phát biểu giữa mối quan hệ giữa con trỏ và mảng một chiều phát biểu nào sau đây sai:

A

Tên của mảng là con trỏ chỉ đến phần tử đầu tiên của mảng

B

Ta có thể thay đổi địa chỉ mà tên mảng trỏ tới

C

Ta có thể duyệt mảng bằng cách sử dụng một biến con trỏ

D

Ta có thể thay đổi giá trị phần tử trong mảng thông qua con trỏ

Trắc nghiệm (tự trả lời)

45

Hãy cho biết giá trị của a và b sau khi thực hiện đoạn chương trình dưới đây

A

a = 5, b = 12;

B

a = 5, b = 5;

C

a = 12, b = 5;

D

a = 12, b = 12;

```
#include <stdio.h>
#include <conio.h>
int a = 5, b = 12;
void swap ( int *a, int *b) {
    int temp = *a; *a = *b; *b = temp;
}
void main(void){
    swap(&a, &b);
    printf("\n a= %d b= %d",a,b);
}
```

Nội dung

- 1 Trắc nghiệm
- 2 Tự luận

Tự luận - Đề quy

1

Thông số hóa
bài toán

- Tổng quát hóa bài toán cụ thể thành bài toán tổng quát.
- Thông số hóa cho bài toán tổng quát
- VD: n trong hàm tính tổng $S(n)$, ...

Tìm thuật giải
tổng quát

- Chia bài toán tổng quát ra thành:
 - Phần không đệ quy.
 - Phần như bài toán trên nhưng kích thước nhỏ hơn.
- VD: $S(n) = S(n - 1) + n$, ...

Tìm các trường hợp
suy biến (neo)

- Các trường hợp suy biến của bài toán.
- Kích thước bài toán trong trường hợp này là nhỏ nhất.
- VD: $S(0) = 0$

Tự luận - Đề quy

1

Viết hàm đệ quy tính tổng sau ($n > 0$):

$$S(n) = 1^2 + 2^2 + 3^2 + \dots + n^2$$

$$S(n) = 1 + \frac{1}{1+2} + \frac{1}{1+2+3} + \dots + \frac{1}{1+2+3+\dots+n},$$

$$S(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n},$$

$$S(n) = 1 + (1+2) + (1+2+3) + \dots + (1+2+3+\dots+n)$$

$$S(n) = 1 + 3 + 5 + \dots + (2 \times n + 1)$$

Tự luận – DS liên kết

2

❖ Duyệt danh sách

- Đếm/Tìm các phần tử của danh sách thỏa điều kiện

```
void DoList(List l)
{
    Node *p;
    p=l.pHead;
    while(p!=NULL)
    {    //Thực hiện phép kiểm tra, tính toán
        p=p->pNext;
    }
}
```


Tự luận – DS liên kết

2

❖ Sắp xếp danh sách

```
void SortList(LIST &l)
{
    Node *p,*q,*min;
    Data temp;
    p=l.pHead;
    while(p!=l.pTail)
    {
        min=p;
        q=p->pNext;
```

```
        while(q!=NULL)
        {
            if(q->Info < min->Info)
                min=q;
            q=q->pNext;
        }
        temp = p->Info;
        p->Info = min->Info;
        min->Info = temp;
        p=p->pNext;
    }
}
```

Tự luận – Tối ưu mã nguồn

3

❖ Các kỹ thuật tối ưu mã nguồn:


- ✓ Tinh chỉnh các biểu thức logic
- ✓ Tinh chỉnh các vòng lặp
- ✓ Tinh chỉnh việc biến đổi dữ liệu
- ✓ Tinh chỉnh các biểu thức
- ✓ Tinh chỉnh dãy lệnh
- ✓ Viết lại mã nguồn bằng ngôn ngữ assembler

Tự luận – Tối ưu mã nguồn

3

❖ Ví dụ:

```
for ( i = 0; i < count; i++ ) {  
    if ( sumType == SUMTYPE_NET ) {  
        netSum = netSum + amount[ i ];  
    }  
    else {  
        grossSum = grossSum + amount[ i ];  
    }  
}
```



```
if ( sumType == SUMTYPE_NET ) {  
    for ( i = 0; i < count; i++ ) {  
        netSum = netSum + amount[ i ];  
    }  
}  
else {  
    for ( i = 0; i < count; i++ ) {  
        grossSum = grossSum + amount[ i ];  
    }  
}
```

Tự luận – Tối ưu mã nguồn

3

❖ Ví dụ:

- Thay thế phép nhân bằng phép cộng
- Thay thế phép lũy thừa bằng phép nhân
- Thay việc tính các hàm lượng giác bằng cách gọi các hàm lượng giác có sẵn
- Sử dụng kiểu dữ liệu có kích thước nhỏ nếu có thể
 - long long int → long, int
 - floating-point → fixed-point, int
 - double-precision → single-precision
- Thay thế phép nhân đôi / chia đôi số nguyên bằng phép bitwise
- Sử dụng hằng số hợp lý
- Tính trước kết quả
- Sử dụng biến trung gian

Câu hỏi ngắn – Cấu trúc DL và STL

Q1

Hỏi đoạn chương trình sau in ra kết quả gì?

```
vector<int> v(50, 1);  
int s = 0;  
for(int i = 0; i < 50; i += 2) {  
    v[i] += v[i+1];  
    s += v[i] + v[i+1];  
}  
cout << s;
```

Câu hỏi ngắn – Cấu trúc DL và STL

Q2

Hỏi đoạn chương trình sau in ra kết quả gì?

```
int a[] = {3, 5, 7, 8, 7, 4};  
sort(a + 1, a + 5);  
for(int i = 0; i < 6; i++)  
    cout << a[i] << " ";
```

Câu hỏi ngắn – Đề quy

Q3

Hãy xác định giá trị của $f(4)$ nếu hàm f được định nghĩa theo kiểu đệ quy như sau:

```
int f(int n) {  
    if (n > 10) return 1;  
    else return f(n*2) + f(n + 1);  
}
```

Tự luận – Tinh chỉnh mã

Q4

Hãy cải tiến để tăng hiệu năng đoạn mã lệnh sau (a , b là hằng số) và giải thích lý do:

```
for (i = 0; i < 100; i++)  
    printf ("%f\n", i*15*cos(a) - sqrt(b));
```


Tự luận – Tinh chỉnh mã

Q5

Hãy tinh chỉnh đoạn mã lệnh sau bằng mọi kỹ thuật và tại mọi vị trí có thể và giải thích lý do (Biết rằng $n > 20$, x, y, z là các số nguyên và các hàm trong đoạn lệnh đã tối ưu).

```
for (int i = 0; i < n; i++) {  
    if (x == z * 64)  
        DoOneThing(i);  
    else if (x == y/32 + z/32)  
        DoSomethingElse(i);  
    else  
        DoAnotherThing(i);  
}
```

Tự luận – Quản lý bộ nhớ

Q6

Cho biết kết quả hiện ra trên màn hình khi thực hiện chương trình dưới đây. Hãy giải thích vì sao lại có kết quả như vậy?

```
#include <stdio.h>
int F(int *x) {
    *x +=1;    return *x;
}
void main(void) {
    int a =5;
    printf("F(a) +F(a) =
%d\n", F(&a)+F(&a));
    a=5;
    printf("2*F(a) = %d\n", 2*F(&a));
}
```

Tự luận – Cấu trúc DL và STL

Q7

Cho một danh sách liên kết đơn quản lý sinh viên với các nút được khai báo như sau:

```
struct node{  
    int mssv;  
    char hoten[30];  
    int namsinh;  
    struct node *next;  
};
```

Hãy viết hàm nhập vào bàn phím số nguyên n là số lượng sinh viên cần quản lý, sau đó hàm tiếp tục yêu cầu nhập thông tin lần lượt từng sinh viên. Yêu cầu dùng kỹ thuật Assertion để đảm bảo n phải là số nguyên dương không quá 200.