

Kỹ thuật lập trình C

Engineering programming in C

Giảng viên phụ trách: thầy Nguyễn Chí Linh

Thông tin chung

- **Số tín chỉ: 2**
 - *Số tiết LT trên lớp:* 30 tiết (15 tuần)
 - *Số tiết tự học:* 60 tiết
- **Nội dung** chủ yếu tập trung vào các phần như: Mô tả thuật toán bằng lưu đồ cho một yêu cầu lập trình cụ thể; Viết mã nguồn ứng với một thuật toán cụ thể bằng ngôn ngữ C/C++ cho ra sản phẩm theo yêu cầu; Tổ chức chương trình (mức độ nhỏ) sao cho dễ bảo trì, có thể mở rộng và tái sử dụng.

Lý thuyết

Bài tập tại lớp, bài tập về nhà và bài tập làm thêm

Bài tập giữa kỳ

Thi
cuối
kỳ



Thông tin giảng viên

- **TS. Nguyễn Chí Linh**
- Bộ môn Vật lý Tin học
- Khoa Vật lý - Vật lý Kỹ thuật
- nclinh@hcmus.edu.vn

Lịch tiếp sinh viên:

- Thời gian: thứ 6 hằng tuần (9g-11g30, 13g30-16g30) .
- Địa điểm: phòng E304 toà nhà E, Trường ĐH Khoa học Tự nhiên

1. Trần Đan Thư, *Giáo trình lập trình C: tập I, II*, NXB Đại Học Quốc Gia TP. HCM,
2. Trần Đan Thư ...[et al], *Kỹ thuật lập trình*, NXB Khoa Học và Kỹ Thuật, 2014
3. Phạm Văn Át và Đỗ Văn Tuấn, *Giáo trình kỹ thuật lập trình C cơ sở và nâng cao*, NXB Thông Tin và Truyền Thông, 2001.

Website

The C++ resources network, www.cplusplus.com

Chuẩn đầu ra

Sau khi tham gia môn học này, sinh viên có thể:

G1	Vận dụng ngôn ngữ lập trình C để tính toán và mô phỏng
G2	Kỹ năng quản lý tài nguyên cá nhân
G3	Kỹ năng nói, trình bày và thảo luận
G4	Xác định vai trò và đánh giá giữa các thành viên trong nhóm
G5	Trung thực và có trách nhiệm

Đánh giá

Bài tập tại lớp (10%)

- Gồm 2 bài tập

Bài tập về nhà (20%)

- Gồm 3 bài tập được nộp và trả kết quả online.
- Chỉ được nộp bài trong khoảng thời gian quy định, nếu nộp trễ sẽ không được tính điểm.

Bài tập giữa kỳ (20%)

- Nộp kết quả qua email cho GV trước buổi trình bày tại lớp.
- GV chỉ định thành viên nhóm trình bày kết quả
- Sinh viên trả lời câu hỏi của giáo viên và của lớp trong 10 phút.

Thi cuối kỳ (50%)

- Tự luận
- Không sử dụng tài liệu, không trao đổi.

- Điểm tối đa bao gồm điểm cộng: 10 điểm.

- **Không đạt:** điểm < 5 , - **Đạt:** điểm ≥ 5

phải học lại học phần này.

0đ

5đ

10đ

Chưa hoàn thành

Hoàn thành

Cấm thi đối với các trường hợp:

- Vắng hơn 4 tiết học.
- Không tuân thủ sự hướng dẫn học tập của GV (tùy mức độ).

Chương 1

Giới thiệu

- Các khái niệm cơ bản
- Mô tả thuật toán
- Sử dụng lưu đồ mô tả thuật toán

TÌM HIỂU

Nội dung tự học

Sv làm theo nhóm: Lập trình là gì? Thuật toán là gì?

1) So sánh ngôn ngữ lập trình C với hợp ngữ và với C#, với matlab;

2) Liệt kê và mô tả ngắn gọn 4 ngôn ngữ lập trình cùng cấp với C (lập trình cấp cao);

3) Tìm hiểu môi trường lập trình (IDE) Dev-C++ và nêu 5 đặc tính (chức năng) của IDE này.

4) Cài đặt Dev-C++ trong môi trường windows xp, 7, 8. Phần 4 được kiểm tra trong buổi thực hành tuần 2.

1

2

3

4

5

6

7

8

9

10

1

2

3

4

5

6

7

8

9

20

Bộ ký tự trong C

- Bộ chữ cái 26 ký tự Latinh **A, B, C, ..., Z,**
- **a, b, c, ..., z**
- Bộ chữ số thập phân : **0, 1, 2, ..., 9**
- Các ký hiệu toán học : **+ - * / = < > ()**
- Các ký tự đặc biệt : **. , : ; [] % \ # \$ ' "**
- Ký tự gạch nối **_** và khoảng trắng **' '**

TÌM HIỂU

Tìm hiểu các
Các kiểu dữ liệu cơ sở ???

- [?] Kiểu số nguyên:
- [?] Kiểu số thực:
- [?] Kiểu luận lý:
- [?] Kiểu ký tự:

1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20

Một ví dụ đơn giản LT-C !!!

```
1
2
3
4
5 /*chao.c*/
6 #include <stdio.h>
7
8
9 void main()
10 {
11     printf("Chao sv VLTH");
12     getch();
13 }
14
15
16
17
18
19
20
```

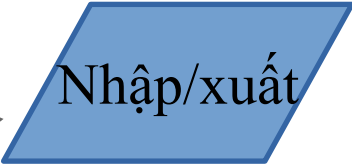
Diễn giải

```
1
2
3
4
5 /*chao.c*/
6 #include <stdio.h> /*Khai bao thu vien*/
7
8
9 void main()
10 {
11     printf("Chao sv VLTH"); /*hien thi len mang hinh*/
12     getch(); /*doi nhap vao 1 ky tu*/
13 }
14
15
16
17
18
19
20
```

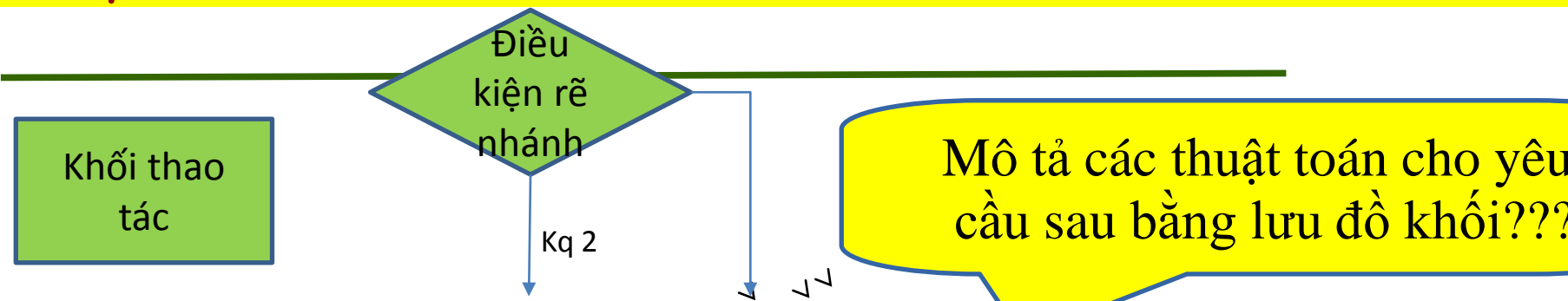
Liên hệ lưu đồ thuật toán ???

```
/*chao.c*/  
#include <stdio.h>
```

```
void main()  
{  
    printf("Chao sv VLTH");  
    getch();  
}
```



1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20



1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20

Cho n là số nguyên dương, x là số thực bậc kỳ. Tính

$$S(n, x) = x - x^2 + x^3 + (-1)^{n+1}x^n$$

Bắt đầu

$$S(n, x) = x - x^3 + x^5 + \dots + (-1)^n x^{2n+1}$$

Nhập/xuất

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

$$S(n) = -x + \frac{x^2}{1+2} - \frac{x^3}{1+2+3} + (-1)^n \frac{x^n}{1+2+3+\dots+n}$$

Kết thúc

Chương 2

Kiểu dữ liệu cơ sở và các phép toán

- Kiểu dữ liệu cơ sở
- Hàm nhập xuất
- Toán hạn, toán tử, biểu thức, lệnh gán

KIỂU SỐ NGUYÊN

Có dấu n bit: $-2^n \rightarrow -2^{n-1} - 1$

Không dấu n bit: $0 \rightarrow 2^n - 1$

1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20

Kiểu (Type)	Độ lớn (Byte)	Miền giá trị (Range)
char	1	-128 -> 127
int	2	-32.768 -> 32.767
short	2	-32.768 -> 32.767
long	4	-2.147.483.648 -> 2.147.483.647
unsigned char	1	0 -> 255
unsigned int	2	0 -> 65.535
unsigned short	2	0 -> 65.535
unsigned long	4	0 -> 4.294.967.295

KIỂU SỐ THỰC

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

Độ chính xác đơn **float**: Gần đúng đến 7 số lẻ.

Độ chính xác kép **double**: Gần đúng đến 19 số lẻ.

Kiểu (Type)	Độ lớn (Byte)	Miền giá trị (Range)
float	4	$3.4 * 10^{-38} \rightarrow 3.4 * 10^{38}$
double	8	$1.7 * 10^{-308} \rightarrow 1.7 * 10^{308}$

Minh họa

$$1711.2016 = 1.7112016 * 10^3$$

KIỂU LUẬN LÝ

C: không tường minh

+ Đúng **true**: giá trị khác **0** (input), trả về **1** (output).

+ Sai **false**: giá trị **0** (input và output).

C++: **bool**

Minh họa

1 (**true**), 0 (**false**), 4.7 (**true**)

$3 > 4$ (**0**, **false**), $5 < 8$ (**1**, **true**)

1

2

3

4

5

6

7

8

9

10

1

2

3

4

5

6

7

8

9

20

KIỂU KÝ TỰ

Tên kiểu: **char**

Miền giá trị: **256** ký tự trong bảng mã **ASCII**

Giá trị: mã **ASCII** của ký tự

Minh họa

Lưu giá trị **65** tương ứng ký tự 'A'

Lưu giá trị **97** tương ứng ký tự 'a'

Gán 'C' tương ứng lưu giá trị **67**

LỆNH GÁN

Gán bởi biểu thức

```
1  Tuan3_2.c
2  =====
3  #include <stdio.h>
4  int a;
5  float b;
6  void main()
7  {
8  /*....*/
9      a = 201;
10     b = 30.09;
11     x=a*a+7*b; // một ví dụ về biểu thức
12     printf("a = %d, b = %f, x = %f\n",a,b,x);
13 }
14
15
16
17
18
19
20
```

LỆNH GÁN

1 Biến có thể được gán bởi

2 1 giá trị

3 1 biến khác

4 1 biểu thức

5
6
7
8 Tuan3_2.c

9 =====

10 **#include** <stdio.h>

1 **int** a;

2 **float** b;

3 **void** main()

4 {

5 /*....*/

6 a = 2016;

7 b = 30.09;

8 x=a*a+7*b; // một ví dụ về biểu thức

9 printf("a = %d, b = %f, x = %f\n",a,b,x);

20 }

MỘT SỐ KHÁI NIỆM

Biểu thức:

là tổ hợp của toán tử tác động lên toán hạng

Toán hạng:

hằng, biến, lời gọi hàm

Toán tử:

+ toán học: +, -, *, /, ++, --, %

+ logic: &&, ||, !

+ so sánh: ==, <, >, <=, >=, !=

+ phẩy: ví dụ, x=(a=2, b=a+3);

+ điều kiện: <biểu thức 1>?<biểu thức 2>:<biểu thức 3>
bt1 đúng thì chọn bt2; bt1 sai thì chọn bt3.

chú ý: toán tử điều kiện có 3 toán hạng => còn gọi là toán tử 3 ngôi.

1

2

3

4

5

6

7

8

9

10

1

2

3

4

5

6

7

8

9

20

THỨ TỰ ƯU TIÊN TOÁN TỬ

1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20

stt	Toán tử	Độ ưu tiên
1	() [] -> .	→
2	! ~ & * ++ --	←
3	* / %	→
4	<< >>	→
5	< <= > >=	→
6	== !=	→
7	&	→
8		→
9	^	→
10	&&	→
11		→
12	?:	←
13	= += -= *= /= %= <<= >>= &= ^= =	←
14	,	←

THỨ TỰ ƯU TIÊN TOÁN TỬ

Viết biểu thức kiểm tra các mệnh đề thỏa yêu cầu
giá trị biểu thức khác 0 cho biết mệnh đề tương ứng là đúng

+ a và b cùng dấu: $(a*b>0)$

+ a bằng b bằng c: $(a==b \ \&\& \ b==c)$

+ $|a| < 1$: $(a>-1 \ \&\& \ a<1)$

Giải thích ý nghĩa biểu thức

+ $a>b?a:x?y:z; \Rightarrow (a>b)?a:(x?y:z)$

+ $a=;b=++a; \Rightarrow a=4, b=4$

+ $x=5;y=x++; \Rightarrow x=6, y=5$

NHẬP XUẤT

Hàm xuất

```
1  int printf(const char *format, ...);
2  =====
3  #include <stdio.h>
4  int a;
5  float b;
6  void main()
7  {
8  /*....*/
9      a = 2016;
10     B = 2016.3009;
1     printf("\nlan 1:\n xuat a = %d\n xuat b = %f",a,b);
2     printf("\nlan 2:\n xuat a = %8d\n xuat b = %11.4f",a,b);
3 }
4
5
6
7
8
9
20
```

NHẬP XUẤT

Ký tự định dạng

1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20

stt	ký tự	ý nghĩa	kiểu
1	%c	ký tự	char
2	%d, %ld	số nguyên	char, int, short, long
3	%f, %lf	số thực	float, double
4	%s	chuỗi ký tự	char[], char *

NHẬP XUẤT

Ký tự điều khiển

1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20

stt	ký tự	ý nghĩa
1	\a	tiếng bip
2	\b	lùi lại 1 bước
3	\n	xuống dòng
4	\t	tab
5	\\	in dấu \
6	\?	in dấu ?
7	\”	in dấu ”

NHẬP XUẤT

Minh họa

```
1  int printf(const char *format, ...);
2  =====
3  /*kieukitu.c*/
4  #include <stdio.h>
5
6  void main()
7  {
8      char c=65, t='a';
9      int i=67;
10     printf("in voi kieu ki tu %cc\n", '%'); // %cc => %c
11     printf("bien char c=65 ung voi gia tri: %c\n", c); // %c => A
12     printf("bien char t='a' ung voi gia tri: %c\n", t); // %c => a
13     printf("bien int i=67 ung voi gia tri: %c\n", i); // %c => C
14     printf("in voi kieu ki tu %cd\n", '%'); // %cd => %d
15     printf("bien char c=65 ung voi gia tri: %d\n", c); // %d => 65
16     printf("bien char t='a' ung voi gia tri: %d\n", t); // %d => 97
17     printf("bien int i=67 ung voi gia tri: %d\n", i); // %d => 67
18     // getch();
19 }
```

NHẬP XUẤT

Hàm nhập

```
1  int scanf(const char * format, ...);
2  =====
3  /*kieukitu.c*/
4  #include <stdio.h>
5
6  int a;
7  float b;
8  void main()
9  {
10     scanf("%d",&a);
11     scanf("%f",&b);/* chú ý không định dạng.*/
12     /*=> scanf("%d%f",&a,&b); */
13     printf("\nlan 1:\n xuat a %d\n xuat b %f",a,b);
14     printf("\nlan 2:\n xuat a %8d\n xuat b %11.4f",a,b);
15 }
16
17
18
19
20
```

THẢO LUẬN

Viết mã nguồn C: đơn giản

1 Nhập năm sinh của một người xuất ra tuổi người đó
2
3

4 Nhập 2 số a và b khác không, xuất ra tổng, hiệu tích thương.
5
6

7 Nhập điểm thi 3 môn toán, lý, hóa của một sinh viên, xuất ra
8 điểm trung bình. *Biết lý hệ số 2, toán và hóa hệ số 1.*
9
10

1 Nhập bán kính đường tròn, xuất ra chu vi và diện tích.
2
3

4 Nhập số lượng và đơn giá một sản phẩm. Tính tiền + thuế
5
6 gtgtmà khách hàng phải trả. *Biết*
7 *tiền chưa thuế = số lượng x đơn giá;*
8 *Thuế = 10% x tiền chưa thuế.*
9
20

Chương 3

Cấu trúc điều khiển

- Khối lệnh
- Cấu trúc rẽ nhánh
- Cấu trúc lặp

KHÔNG LỆNH

Tuan4_1.c

=====

//Cac lenh

{

/*....*/

//Cac lenh

}

//Cac lenh

1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20

KHỐI LỆNH

Quy ước (trong môn học)

Biến chung: Biến khai báo bên ngoài khối lệnh.

Biến riêng: Biến khai báo bên trong khối lệnh.

Biến riêng **trùng tên** biến chung thì việc sử dụng biến này bên trong khối lệnh không làm thay đổi biến chung khi ra khỏi khối lệnh.

Nếu biến chung **không trùng tên** với biến riêng, việc sử dụng biến chung trong khối lệnh có thể làm thay đổi giá trị biến này.

KHỐI LỆNH

Minh họa

```
1  #include <stdio.h>
2
3  void main()
4  {
5      int a=2015, b=304; // khai báo và gán giá trị cho a và b
6      printf("truc khoi lenh con\n"); // in dòng chữ và xuống hàng
7      printf("a = %d\n",a); // in a và xuống hàng
8      printf("b = %d\n",b); // in b và xuống hàng
9      {                               //bắt đầu 1 khối lệnh con (klcon)
10         int b=205; // biến riêng b của klcon trùng tên với biến b chung.
11         a=2016; // klcon sử dụng biến a chung
12         printf("trong khoi lenh con\n");
13         printf("dat/gan lai gia tri a, khai bao bien b\n");
14         printf("a = %d\n",a);
15         printf("b = %d\n",b);//biến b được hiểu là biến b riêng của khối lệnh
16     } //kết thúc khối lệnh con, biến b riêng không tồn tại bên ngoài klcon
17     printf("ra ngoai khoi lenh con\n");
18     printf("a = %d\n",a); //biến a bị thay đổi thông qua khối lệnh.
19     printf("b = %d\n",b); //biến b ở đây là của chung.
20 }
```

KHÔI LỆNH

Minh họa

//Gọi sv giải thích..

```
1  /*bienchung.c*/
2  #include <stdio.h>
3
4  int a=17; //hạn chế sử dụng, hoặc được sự thống nhất của trường nhóm lập trình.
5  void main()
6  {
7      printf("khai bao a truooc main()\n");
8      printf("a = %d\n",a);
9      {
10         int a = 1983;
1         printf("trong khoi lenh: khai bao a\n");
2         printf("a = %d\n",a);
3     }
4     printf("ra ngoai khoi lenh\n");
5     printf("a = %d\n",a);
6     // getch();
7 }
8
9
20
```

CẤU TRÚC RỄ NHÁNH

if, if..else

//Gọi sv vẽ sơ đồ khối ..

```
1  /*cuctriif.c*/
2  #include <stdio.h>
3
4  void main()
5  {
6      int a,b;
7      int max,min;
8      printf("nhap a = ");
9      scanf("%d",&a);
10     printf("nhap b = ");
1     scanf("%d",&b);
2     max=a; min=b;
3     if (b>a) {
4         max=b; min=a;
5     }
6     printf("max = %d\n",max);
7     printf("min = %d\n",min);
8     getch();
9 }
20
```

CẤU TRÚC RỄ NHÁNH

if, if..else

//Gọi sv vẽ sơ đồ khối ..

```
1  /*cuctriif.c*/
2  #include <stdio.h>
3
4  void main()
5  {
6      int a,b;
7      int max,min;
8      printf("nhap a = ");
9      scanf("%d",&a);
10     printf("nhap b = ");
1     scanf("%d",&b);
2     if (b>a) {
3         max=b; min=a;
4     } else {
5         max=a; min=b;
6     }
7     printf("max = %d\n",max);
8     printf("min = %d\n",min);
9     getch();
20 }
```

CẤU TRÚC RỄ NHÁNH

if, if .. else

Hoạt động nhóm

Viết CT sử dụng cấu trúc if , if else giải biện luận phương trình

1) Bậc nhất: $ax + b = 0$

2) Bậc hai: $ax^2 + bx + c = 0$ (bt)

CẤU TRÚC RỄ NHÁNH

switch..case

```
1
2
3  switch (bieu_thuc) {
4      case giatri1:
5          khoilenh1
6          break;
7      case giatri2:
8          khoilenh2
9          break;
10     case giatriN:
11         khoilenhN
12         break;
13     default:
14         khoilenh0
15 }
16
17
18
19
20
```

CẤU TRÚC RỄ NHÁNH

switch..case

//Gọi sv giải thích..

```
1
2  #include <stdio.h>
3  void main()
4  {      int n;
5         char* chuso;
6         printf("nhap vao 1 so nguyen: "); scanf("%d",&n);
7         switch (n){
8             case 0: chuso="so khong";break;
9             case 1: chuso="so mot";break;
10            case 2: chuso="so hai";break;
1             case 3: chuso="so ba";break;
2             case 4: chuso="so bon";break;
3             case 5: case 6: case 7: case 8: case 9: case 10:
4                 chuso="ah, lon hon 4 nho hon 11";break;
5             default:
6                 if (n<0) chuso="chua duoc huan luyen voi so am";
7                 else chuso = "chua duoc huan luyen voi so lon hon 10";
8         }
9         printf("\ncau tra loi la: %s\n",chuso);
20 }
```

CẤU TRÚC RỄ NHÁNH

Thảo luận

1 Hoạt động nhóm

2
3
4
5
6
7
8
9 Viết CT nhập vào ngày tháng năm, xuất ra quý trong năm (**bt**)

10
1
2
3
4
5
6
7
8
9
20

CẤU TRÚC RỄ NHÁNH

while

//Cac lenh ke tiep

while(bt_dieu_kien)
khoi lenh A

//lenh ke tiep

CẤU TRÚC RỄ NHÁNH

while

ví dụ:

Viết chương trình xuất ra 10 lần dòng chữ:

“chep phat: di tre tu vao lop, khong lam on ao”

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    int k=1;
```

```
    while (k<=10){
```

```
        printf("%d, chep phat: di tre tu vao lop, khong lam on ao\n",k);
```

```
        k++;
```

```
    }
```

```
}
```

CẤU TRÚC RỄ NHÁNH

while

Thảo luận:

Dùng cấu trúc lặp while, viết chương trình tính

1, Tổng N số nguyên dương đầu tiên.

2, Tích N số nguyên dương đầu tiên (N!)

1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20

CẤU TRÚC LẶP

for

Cấu trúc

```
////////////////////////////////////
```

```
for (bieu_thuc_1; bieu_thuc_2;bieu_thuc_3)
```

```
    khoi_lenh_A
```

```
//cac lenh_khac
```

```
////////////////////////////////////
```

ví dụ:

in ra 10 lần dòng chữ:

“phai viet duoc thuat toan truoc”

“roi moi viet code”

CẤU TRÚC LẶP

for

```
1  
2  
3  
4  
5 #include <stdio.h>
```

```
6  
7 void main()
```

```
8 {
```

```
9     int i;
```

```
10    for (i=0;i<10;i++) {
```

```
1    printf("%d, phai viet duoc thuat toan truoac\n",i);
```

```
2    printf("roi moi viet code,\n");
```

```
3  
4    }
```

```
5 }
```

```
6
```

```
7
```

```
8
```

```
9
```

```
20
```


LUYỆN TẬP

1 1, **(bt)** Viết chương trình xuất ra tam giác vuông cân dưới bên
2
3 trái bằng ký tự '*' thỏa

4
5 2, **(bt)** Viết chương trình cài đặt 5 bài tập trong phần viết sơ đồ
6
7 khối mà nhóm được giao trong tuần 1-2.
8

9
10 3, **(bt)** Viết chương trình xuất ra tam giác vuông cân bằng ký
1
2 tự '*' thỏa:

3 1, trên bên trái

4 2, dưới bên phải

5 3, trên bên phải.
6
7
8
9
20

ÔN TẬP

Vẽ sơ đồ khối các cấu trúc điều khiển

1

2

3

4

5

6

7

8

9

10

1

2

3

4

5

6

7

8

9

20

Chương 4

Con trỏ và hàm

- Toán tử *
- Biến con trỏ
- Hàm

TOÁN TỬ *

=====

1 Năm giữa 2 biến, 2 số có nghĩa là nhân,

2

3 Kết hợp với dấu bằng, $x *= a$: tương đương $x = x * a$;

4

5 =====

6

7 Các trường hợp có dạng sau đây thì sao?

8

9

10 **int *x;**

1

2

3 ==>

4

5

6

7 ***x;**

8

9 ==>

10

20 =====

TOÁN TỬ *

=====

1 Năm giữa 2 biến, 2 số có nghĩa là nhân,

2
3 Kết hợp với dấu bằng, $x *= a$: tương đương $x = x * a$;

=====

6
7 Các trường hợp có dạng sau đây thì:

8
9
10 **int *x;** //khai báo biến x chứa địa_chỉ_vùng_nhớ mà tại đó lưu 1
1 giá_trị_kiểu_nguyên.

2
3 ==> **<kiểu_dl> * x;** // khai báo x là biến địa chỉ mà ở đó nhớ 1
4 giá_trị_kiểu_tương_ứng;

5
6
7 ***x;** //trả về giá trị nguyên tại địa chỉ x;

8 ==> toán tử * truy xuất “nội dung” tại địa chỉ x;

=====

20

TOÁN TỬ *

Hỏi đáp

Diễn giải chi tiết CT sau

```
1
2
3
4  #include<stdio.h>
5  void main()
6  {
7      //      các khai báo;
8      int a, b, c;// khai báo biến a, b kiểu số nguyên
9      int *x;// khai báo biến x chứa địa chỉ của 1 kiểu số nguyên
10     int *y;// khai báo biến y chứa địa chỉ của 1 kiểu số nguyên
1     a = 5;// đặt a bằng 5
2     b = 10;// đặt b bằng 10
3     x = &c;//đặt x bằng địa chỉ của c
4     *x = b;//đặt nội dung tại x (địa chỉ x) bằng b
5     y = &a;//đặt y bằng địa chỉ của a
6     //      các lệnh;
7 }
```

TOÁN TỬ *

Hỏi đáp

Diễn giải chi tiết CT sau

```
1  /*toantusao.c*/
2  #include<stdio.h>
3
4  void main()
5  {
6      int a, b, c;
7      int *x;
8      int *y;
9      a=5; b=10;
10     printf("a = %d, b=%d\n",a,b);
1     x = &c; //phụ thuộc IDE, hđ hành
2     *x = a; printf("gan *x = a, ");
3     printf("=>a =%d, b = %d\n",a,b);
4     y = &b; printf("gan *y = &b\n");
5     *y = 20; printf("gan *y = 20\n");
6     printf("=>a = %d, b = %d\n",a,b);
7     b = 40; printf("gan b = 40\n");
8     printf("=>a = %d, b = %d\n",a,b);
9     printf("=>*y = %d\n",*y);
20 }
```

Kết quả

```
a = 5, b = 10
gan *x = a, =>a =5, b=10
gan *y = &b
gan *y = 20
=>a = 5, b = 20
gan b = 40
=>a = 5, b = 40
=>*y = 40
```

TOÁN TỬ *

Ôn tập

Giả sử khai báo

1 /*giasu.c*/

2

3 int a;

4 int *x=&a;

5

6 Thì:

7

8

9 x; truy xuất gì (địa chỉ hay giá trị)

10 a; truy xuất gì

1

2 *x; truy xuất gì

3 &a; truy xuất gì

4

5

6

7

8

9

20

HÀM

Khái niệm

Là một đoạn chương trình

Có tên, đầu vào, đầu ra

Thực hiện 1 công việc nào đó cho chương trình chính

Thực hiện 1 công việc nào đó cho hàm khác

=> Có tính tái sử dụng
Thuận tiện trong việc sửa lỗi và cải tiến chương trình.

1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20

HÀM

Minh họa

```
1  int a;//biến toàn cục, như sử dụng biến với khối lệnh
2  int hama()
3  {
4      int a;//==> biến a trong hama() được hiểu là biến cục bộ
5      //    các lệnh;
6  }
7
8  void hamb()
9  {
10     int b;//=>biến riêng của hamb()
11     // các lệnh;
12 }
13
14 void main()
15 {
16     int b;//biến riêng của hàm main()
17     //    các lệnh;
18 }
19
20
```

HÀM

Các thông tin cần thiết để xác định hàm

1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20

Thực hiện công việc gì (chức năng)

=> Đặt tên gọi nhớ

Có thuật toán, cài đặt cụ thể.

Đầu vào, đầu ra ?

đầu vào 1----->/////////////////
đầu vào 2----->// //----->đầu ra (nếu có)
đầu vào 3 ----->/////////////////

HÀM

Sinh viên làm cá nhân

Viết cú pháp định nghĩa 1 hàm (giải thích đầy đủ)

1

2

3

4

5

6

7

8

9

10

1

2

3

4

5

6

7

8

9

20

MINH HỌA

Các hàm cộng trừ nhân chia 2 số nguyên

Trả giá trị về cho hàm

```
1
2  int tong(int x, int y) {
3      int kq;
4      kq = x+y;
5      return kq;
6  }
7  int tru(int x, int y) {
8      return x-y;
9  }
10
1  int nhan(int x, int y) {
2      return x*y;
3  }
4  float chia(int x, int y) {
5      if (y!=0) return x/y;
6      else {
7          printf("chia 0, khong xac dinh");
8          return 0;
9      }
20 }
```

MINH HỌA

Các hàm cộng trừ nhân chia 2 số nguyên

Trả giá trị về cho biến

1 void tong(int x, int y, int *kq)**2 {****3 *kq = x+y;****4 }****6 void tru(int x, int y, int *kq)****7 {****8 *kq = x-y;****9 }****1 void nhan(int x, int y, int *kq)****2 {****3 *kq = x*y;****4 }****6 void chia(int x, int y, float *kq)****7 {****8 if (y!=0) *kq = x/y;****9 else printf("chia 0, khong xac dinh");****20 }**

MINH HỌA

Cài đặt cụ thể

/*pheptoa.c*/

```
1  #include<stdio.h>
2
3  void tong(int x, int y, int *kq) {
4      *kq = x+y; }
5  void tru(int x, int y, int *kq) {
6      *kq = x-y; }
7  void nhan(int x, int y, int *kq) {
8      *kq = x*y; }
9  void chia(int x, int y, float *kq) {
10     if (y!=0) *kq = x/y;
1     else printf("chia 0, khong xac dinh"); }
2  void main()
3  {
4     int x, y;
5     int kq_tong, kq_nhan, kq_tru;
6     float kq_chia;
7     printf("nhap x = "); scanf("%d",&x);
8     printf("nhap y = "); scanf("%d",&y);
9     tong(x,y,&kq_tong);
10    printf("tong la: %d\n",kq_tong);
20 }
```

QUẢN LÝ HÀM

Thông thường để dễ quản lý ctrình chính và chương trình con người lập trình thường viết chương trình con bên dưới hàm main() hoặc ở file khác.

Viết lại cấu trúc trên theo tuần tự:

- + Khai báo hàm nguyên mẫu

- + Viết hàm main()

- +Viết hàm sau hàm main()

QUẢN LÝ HÀM

```
1  #include <stdio.h>
2  void tong(int x, int y, int *kq);
3  void tru(int x, int y, int *kq);
4  void nhan(int x, int y, int *kq);
5  void chia(int x, int y, float *kq);
6
7  void main()
8  {
9      int x, y, kq_tong, kq_nhan, kq_tru;
10     float kq_chia;
11     printf("nhap x = "); scanf("%d",&x);
12     printf("nhap y = "); scanf("%d",&y);
13     tong(x,y,&kq_tong);
14     printf("tong la: %d\n",kq_tong);
15 }
16 void tong(int x, int y, int *kq) { *kq = x+y; }
17 void tru(int x, int y, int *kq) { *kq = x-y; }
18 void nhan(int x, int y, int *kq) { *kq = x*y; }
19 void chia(int x, int y, float *kq) {
20     if (y!=0) *kq = x/y;
21     else printf("chia 0, khong xac dinh"); }
```

LUYỆN TẬP

Thảo luận

1 Trình bày thuật giải (các bước thuật toán) định hướng sử dụng hàm,
2 Sử dụng kỹ thuật lập trình hàm viết chương trình
3 theo yêu cầu:
4
5 =====

6
7 Tìm tất cả các số nguyên tố dương $\leq n$
8

9
10
1
2
3
4
5
6 =====

7
8
9
20

LUYỆN TẬP

Thảo luận

1 Trình bày thuật giải (các bước thuật toán) định hướng sử dụng hàm,
2 Sử dụng kỹ thuật lập trình hàm viết chương trình
3 theo yêu cầu:
4

5
6 =====

7
8 1, Giải phương trình bậc 1: $ax + b = 0$
9

10 2, Giải phương trình bậc 2: $ax^2 + bx + c = 0$
1

2
3 3, Giải phương trình trùng phương: $ax^4 + bx^2 + c = 0$
4

5 =====

6 Hướng dẫn:
7 yêu cầu 2 có sử dụng hàm của yêu cầu 1,
8 yêu cầu 3 có sử dụng hàm của yêu cầu 1 và 2.
9
20

Chương 5

Mảng dữ liệu

- Khai báo và sử dụng
- Truyền mảng cho hàm
- Kỹ thuật xử lý cơ bản

ĐẶT VẤN ĐỀ

1 khai báo 3 số nguyên:

2 **int sn1, sn2, sn3;**

3
4 khai báo 10 số nguyên:

5 **int sn1, sn2, sn3, sn4, sn5, sn6, sn7, sn8, sn9, sn10;**

6
7 khai báo 51 số nguyên, thêm 51 số nguyên nữa: ...

8 (quản lý số tín chỉ đã đăng ký, stc đã đạt của 51 sv vlth)

9
10 => khai báo 2 dãy số nguyên n số

1
2
3 Giải pháp:

4 (trong hầu hết các ngôn ngữ lập trình cấp cao)

5 sử dụng khai báo dãy (mảng)

6
7
8
9
10
20 ===> **mảng là gì? khai báo sử dụng ra sao?**

KHÁI NIỆM

Mảng là kiểu dữ liệu có cấu trúc
chứa tập các phần tử có cùng kiểu.

//mỗi phần tử mảng thỏa tính chất của một biến của kiểu được khai báo:
+ Có thể thay đổi giá trị (gán lại)
+ Có thể dùng trong biểu thức.

Mảng một chiều biểu diễn dãy của nhiều phần tử cùng kiểu.

=====

Thảo luận

Cú pháp khai báo mảng?

Cú pháp truy xuất các phần tử trong mảng?

KHÁI NIỆM

Khái báo mảng

1 **int** mang1d[10]; //khai báo 10 phần tử kiểu nguyên

2
3 Phân biệt các cách sử dụng '**[1_so_nguyen]**' đối với mảng.

4
5 **int** a[10]; //khai báo mảng **a** có 10 phần tử

6 **int** x,y;

7
8 **a[0]**; // truy xuất phần tử “0” (thứ nhất) của mảng **a**

9 **a[4]**; // truy xuất phần tử “4” (thứ năm) của mảng **a**

10 **a[9]**; // truy xuất phần tử “9” (cuối) của mảng **a**

1 ...

2 **x = 2; y = 3;**

3 **a[x+y]**; //truy xuất phần tử “giá trị biểu thức” của mảng **a**

4 //giá trị biểu thức **x+y** là **5** => truy xuất phần tử “5” của mảng **a**

5
6 ghi chú: với $0 \leq i < \text{số phần tử của mảng } a$ thì

7 **a[i]**; truy xuất phần tử “**i**” của mảng **a**

8

9

20

KHÁI NIỆM

Khái báo mảng

```
1  #define ssvk13 51
2
3  //int tcdk[51]; //khai báo 51 phần tử kiểu nguyên
4  //int tctl[51];
5  int tcdk[ssvk13]; //khai báo 51 phần tử kiểu nguyên
6  int tctl[ssvk13];
7
8
9
10
```

1
2
3
4
5
6
7
8
9
20

KHÁI NIỆM

Khởi tạo giá trị cho mảng

```
1      int manga[5] = {1, 4, 6, 3, 2};
```

Chú giải

Khai báo **manga** có 5 phần tử

Khởi gán giá trị cho 5 phần tử lần lượt là

```
manga[0] = 1
```

```
manga[1] = 4
```

```
manga[2] = 6
```

```
manga[3] = 3
```

```
manga[4] = 2
```

KHÁI NIỆM

Khởi tạo giá trị cho mảng

```
// int manga[5] = {1, 4, 6, 3, 2};  
int mangb[5] = {9, 6, 1};
```

Chú giải

Khai báo **mangb** có 5 phần tử

Khởi gán giá trị cho 5 phần tử lần lượt là

```
mangb[0] = 9  
mangb[1] = 6  
mangb[2] = 1  
mangb[3] = 0  
mangb[4] = 0
```

KHÁI NIỆM

Khởi tạo giá trị cho mảng

```
1 //      int manga[5] = {1, 4, 6, 3, 2};  
2 //      int mangb[5] = {9, 6, 1};  
3      int mangc[5] = {0};  
4  
5
```

Chú giải

Khai báo **mangc** có 5 phần tử

Khởi gán giá trị cho 5 phần tử lần lượt là

```
      mangc[0] = 0  
      mangc[1] = 0  
      mangc[2] = 0  
      mangc[3] = 0  
      mangc[4] = 0
```

KHÁI NIỆM

Khởi tạo giá trị cho mảng

```
1 //      int manga[5] = {1, 4, 6, 3, 2};  
2 //      int mangb[5] = {9, 6, 1};  
3 //      int mangc[5] = {0};  
4      int mangd[] = {4, 19, 10};  
5
```

Chú giải

Khai báo **mangd**

Khởi gán giá trị cho 3 phần tử lần lượt là

```
      mangd[0] = 4  
      mangd[1] = 19  
      mangd[2] = 10
```

Số phần tử của **mangd** được tự động cập nhật là 3

MINH HỌA

```
/*mang.c*/
```

```
1  #include <stdio.h>
2
3  void main()
4  {
5      #define ssvk13 51
6      int tctl[ssvk13];
7      int tcdk[ssvk13] = {88,88,88,50}; //cac phan tu khac bang 0
8      int dotuoi[] = {20,21,22};
9      int i, tam;
10
11     tctl[0] = 84; tctl[1] = 85; tctl[2] = 88; tctl[3] = 44;
12     //cac phan tu khac co the nhan gia tri nguyen bat ky
13     tam = sizeof(tctl)/sizeof(tctl[0]);
14     printf("\n do dai toi da cua mang tctl la: %d\n",tam);
15     tam = sizeof(dotuoi)/sizeof(dotuoi[0]);
16     printf("\n do dai toi da cua mang dotuoi la: %d\n",tam);
17     for (i=0;i<10;i++){
18         printf("tcdk[%d] = %d\n",i,tcdk[i]);
19         printf("tctl[%d] = %d\n",i,tctl[i]);
20     }
```

MINH HỌA

`/*mang.c*/`

```
1  do dai toi da cua mang tctl la: 51
2
3  do dai toi da cua mang dotuoi la: 3
4  tcdk[0] = 88
5  tctl[0] = 84
6  tcdk[1] = 88
7  tctl[1] = 85
8  tcdk[2] = 88
9  tctl[2] = 88
10 tcdk[3] = 50
1  tctl[3] = 44
2  tcdk[4] = 0
3  tctl[4] = 1
4  tcdk[5] = 0
5  tctl[5] = 32767
6  tcdk[6] = 0
7  tctl[6] = -1390276248
8  tcdk[7] = 0
9  tctl[7] = 54
20 tcdk[8] = 0
21 tctl[8] = 0
22 tcdk[9] = 0
23 tctl[9] = 0
```

TRUYỀN MẢNG CHO HÀM

```
1 //khai bao thu vien
2 void nhapmang(int a[51], int *n); //tham kiểu mảng có tối đa 51 phần tử
3 void xuatmang(int a[], int n); //có thể bỏ chỉ số “số phần tử tối đa”
4
5 void main()
6 {
7     int a[51], n;
8     // các xử lý bên trong hàm có thể làm thay đổi nội dung của mảng
9     nhapmang(a, &n);
10    xuatmang(a, n);
11 }
12 //dinh nghĩa ham nhapmang(int a[51], int *n) {...}
13 //dinh nghĩa ham xuatmang(int a[], int n) {...}
14
15
16
17
18
19
20
```

TRUYỀN MẢNG CHO HÀM

```
1 //khai bao thu vien
```

```
2 void nhapmang(int a[51], int *n); //tham kiểu mảng có tối đa 51 phần tử
```

```
3 void xuatmang(int a[], int n); //có thể bỏ chỉ số “số phần tử tối đa”
```

4
5
6
7
8 Trong thực tế, số lượng phần tử cần dùng không cố định. Do đó, khi viết
9 chương trình, người lập trình cần ước lượng số phần tử tối đa đủ lớn ứng với
10 nhu cầu tính toán, xử lý đang xét.

1
2
3 Số phần tử thực sự n của mảng không được lớn hơn số phần tử tối đa N

SV LÀM THEO NHÓM

Cài đặt chương trình nhập vào 1 mảng, thực hiện

- + Xuất số các số nguyên tố
- + Xuất số các số chẵn, lẻ
- + Sắp xếp không giảm, xuất mảng sau khi sắp xếp
- + Sắp xếp không tăng, xuất mảng sau khi sắp xếp

1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
20

SV LÀM THEO NHÓM

Cài đặt chương trình nhập vào 1 mảng, thực hiện

- + Xuất số các số nguyên tố
- + Xuất số các số chẵn, lẻ
- + Sắp xếp không giảm, xuất mảng sau khi sắp xếp
- + Sắp xếp không tăng, xuất mảng sau khi sắp xếp

Chia nhóm như sau:

0, nhập xuất mảng, chép mảng (gv)

1, a) hàm kiểm tra số nguyên tố ?

b) hoán vị 2 số

2, a) so sánh 2 số là tăng, giảm, bằng ?

b) 1 số là chẵn, lẻ ?

3, a) đếm số chẵn, lẻ trong mảng,

b) đếm số nguyên tố trong mảng, có bao nhiêu hay ko có?

4, Sắp xếp mảng không giảm

5, Sắp xếp mảng không tăng

6, Cài đặt chương trình

GIỚI THIỆU

Mảng 2D biểu diễn cho một ma trận $m \times n$

Trong đó, các phần tử có cùng kiểu.

m : biểu diễn số dòng,

n : biểu diễn số cột,

KHAI BÁO

1 =====

2
3 <kiểu_cơ_sở> <tên_biến>[<N1>][<N2>;

4
5
6 =====

7 N1,N2:Số lượng phần tử mỗi chiều

8
9 ví dụ

10
1 int mang2d1[5][5];

2 int mang2d2[2][5];

3 int mang2d3[5][2];

4
5
6 =====

7 Truy xuất

8 mang2d1[0][0]; mang2d1[1][4]; mang2d2[0][2]; mang2d2[1][4];

9 mang2d2[0][0]; mang2d3[1][1]; mang2d3[4][1]; mang2d3[0][0];

20

KHAI BÁO

```
1  =====  
2  
3  
4  typedef <kiểu_cơ_sở> <tên_kiểu_mới>[<N1>][<N2>;
```

```
5  
6  =====  
7  N1,N2:Số lượng phần tử mỗi chiều
```

9 ví dụ

```
10  
1  typedef int mang5x5[5][5];  
2  typedef int mang10x5[10][5];  
3  mang5x5 a; mang10x5 b;
```

```
4  
5  
6  =====  
7  Truy xuất  
8  a[0][0]; a[1][4]; a[4][4];  
9  b[0][0]; b[1][2]; b[9][4];
```

20

GÁN DỮ LIỆU

```
1  #include <stdio.h>
2  #define N 5
3  typedef int mang2d[N][N];
4
5  void main()
6  {    mang2d a, b;
7      int i, j;
8
9      for (i=0;i<5;i++) {
10         for (j=0;j<5;j++) {
1             a[i][j] = i*i + j*j;
2         }
3     }
4     //Gán 2 mảng: bắt buộc gán từng phần tử.
5     for (i=0;i<5;i++) {
6         for (j=0;j<5;j++) {
7             b[i][j] = a[i][j];
8         }
9     }
20 }
```

ĐỔI SỐ CỦA HÀM

Địa chỉ phần tử đầu tiên của mảng được truyền cho hàm

Mảng có thể thay đổi sau khi truyền cho hàm

Khai báo đổi số mảng như khái báo mảng

=====

```
#include <stdio.h>
```

```
#define N 5
```

```
typedef int mang2d[N][N];
```

```
void nhapmang2d(mang2d a);
```

```
void main()
```

```
{
```

```
    mang2d b;
```

```
    nhapmang2d(b);
```

```
}
```

```
void nhapmang2d(mang2d a) {
```

```
//cac lenh tuong ung
```

```
}
```

ĐỔI SỐ CỦA HÀM

Địa chỉ phần tử đầu tiên của mảng được truyền cho hàm

Mảng có thể thay đổi sau khi truyền cho hàm

Khai báo đổi số mảng như khái báo mảng

=====

```
#include <stdio.h>
```

```
#define N 5
```

```
typedef int mang2d[N][N];
```

```
void nhapmang2d(mang2d a);
```

```
void main()
```

```
{
```

```
    mang2d b;
```

```
    nhapmang2d(b);
```

```
}
```

```
void nhapmang2d(int a[N][N]) {
```

```
//cac lenh tuong ung
```

```
}
```


ĐỔI SỐ CỦA HÀM

Địa chỉ phần tử đầu tiên của mảng được truyền cho hàm

Mảng có thể thay đổi sau khi truyền cho hàm

Khai báo đổi số mảng như khái báo mảng

Có thể bỏ chỉ số phần tử thứ 2

=====

```
#include <stdio.h>
```

```
#define N 5
```

```
typedef int mang2d[N][N];
```

```
//void nhapmang2d(mang2d a);
```

```
void nhapmang2d(int a[][N]);
```

```
void main()
```

```
{
```

```
    mang2d b;
```

```
    nhapmang2d(b);
```

```
}
```

```
void nhapmang2d(int a[N][N]) {
```

```
    //cac lenh tuong ung
```

```
}
```

ĐỔI SỐ CỦA HÀM

Địa chỉ phần tử đầu tiên của mảng được truyền cho hàm

Mảng có thể thay đổi sau khi truyền cho hàm

Khai báo đổi số mảng như khái báo mảng

Có thể bỏ chỉ số phần tử thứ 2

=====

```
#include <stdio.h>
```

```
#define N 5
```

```
typedef int mang2d[N][N];
```

```
//void nhapmang2d(mang2d a);
```

```
void nhapmang2d(int a[][N]);
```

```
void main()
```

```
{
```

```
    mang2d b;
```

```
    nhapmang2d(b);
```

```
}
```

```
void nhapmang2d(mang2d a) {
```

```
    //cac lenh tuong ung
```

```
}
```

ĐỔI SỐ CỦA HÀM

Viết chương trình xử lý ma trận A(mxn) theo từng yêu cầu sau

1) Chép dòng k vào dòng h

void chepDong(**int** A[][N], **int** m, **int** n, **int** k, **int** h);

2) Chép cột k vào cột h

void chepcot(**int** A[][N], **int** m, **int** n, **int** k, **int** h);

3) Hoán đổi dòng k và h

void hoandoiDong(**int** A[][N], **int** m, **int** n, **int** k, **int** h);

4) Hoán đổi cột k và h

void hoandoicot(**int** A[][N], **int** m, **int** n, **int** k, **int** h);

=====

Trong đó, $m \leq M$, $n \leq N$

Chương 6

Chuỗi

- Khai báo và sử dụng
- Các hàm thao tác với chuỗi
- Chuỗi trong C++

KHÁI NIỆM

Chuỗi được hiểu là 1 dãy các ký tự //ngôn ngữ C

```
1  #include <string.h>
2  char chuoi1[20];//khai báo chuỗi
3
4  char chuoi2[40] = "Ho va Ten";// khai báo và khởi gán
5  char chuoi3[100];
6  char chuoi4[100];
7
8  scanf("%s",chuoi1);// giả sử nhập: Nguyen Van A
9  => chuoi1 == "Nguyen"
10 //chuỗi nhập với scanf kết thúc bằng khoảng trắng
1
2
3      fgets(chuoi3,100,stdin);// giả sử nhập: Nguyen Van A
4  => chuoi3 == "Nguyen Van A"
5
6
7  strcpy(chuoi4,"Nguyen Van Anh B");//cách gán nội dung cho chuỗi
8  => chuoi4 == "Nguyen Van Anh B"
9
20
```

KHÁI NIỆM

Chuỗi được hiểu là 1 dãy các ký tự // ngôn ngữ C

```
1 #include <string.h>
2 char chuoi1[20]; // khai báo chuỗi
3 char chuoi2[40] = "Ho va Ten"; // khai báo và khởi gán
4 char chuoi3[100];
5 char chuoi4[100];
6
7
8 strcpy(chuoi4, "Nguyen Van Anh B"); // cách gán nội dung cho chuỗi
9
10 => chuoi4 == "Nguyen Van Anh B"
```

Chuỗi luôn được kết thúc bằng ký tự '\0' (null)

```
1
2
3
4 strlen(chuoi); // truy xuất độ dài (số ký tự) thực sự của chuỗi
5 // có thể bao gồm "ký tự enter", kiểm tra với fgets
6
7
8
9
10
```

THẢO LUẬN

Cài đặt chương trình nhập vào 1 chuỗi là 1 cụm từ, 1 câu

1
2
3 1, Đếm số từ

4
5
6 2, Đếm số lần xuất hiện của ký tự a

7
8
9 3, Đếm số khoảng trắng, khử tất cả các khoảng trắng

10
1
2 4, Xuất ra các kết quả đếm.

3
4
5 5, Xuất ra chuỗi sau khi khử khoảng trắng

LUYỆN TẬP

Bài tập giữa kỳ 1

=====

Nhập vào 2 đa thức biến x dạng chuỗi.

Xuất ra tổng, hiệu 2 đa thức,

Xuất ra tích 2 đa thức,

*)Xuất ra thương giữa 2 đa thức và phần dư của phép chia.

**)Nhập vào 2 đa thức dạng hỗn hợp công trừ nhân chia đa thức có sử dụng dấu ngoặc đơn '()'

Xuất ra tổng, hiệu 2 đa thức,

Xuất ra tích 2 đa thức,

Xuất ra thương giữa 2 đa thức và phần dư của phép chia.

=====

Ghi chú: giới hạn hệ số và số mũ chỉ có kiểu nguyên

dạng đa thức: x^2+1 , x^2+1 , $x+x^3-1+x$, ...

LUYỆN TẬP

Bài tập giữa kỳ 1

Yêu cầu về TH:

0, Nhật ký làm việc (**nhóm trưởng báo cáo cho gv theo tuần**)

Bảng phân công công việc của thành viên nhóm: nhóm trưởng làm gì, thư ký làm gì, từng thành viên còn lại làm gì,..theo tuần

CV nào đạt, cv nào không đạt

CV việc kế tiếp cần làm ==> Bảng phân công CV tuần kế tiếp.

1, Cài đặt hàm bằng ngôn ngữ C ứng với mỗi giải thuật.

Vượt qua được các yêu cầu kiểm tra của gv và sv khác.

2, Cài đặt mã nguồn theo yêu cầu đề bài.

Vượt qua được các yêu cầu kiểm tra của gv và sv khác.

=====

Ghi chú: giới hạn hệ số và số mũ chỉ có kiểu nguyên

dạng đa thức: x^2+1 , x^2+1 , $x+x^3-1+x$, ...

Chương 7

Kiểu dữ liệu cấu trúc

- Đặt vấn đề
- Khai báo và sử dụng
- Minh họa

KHÁI NIỆM

Gồm nhiều thành phần, mỗi thành phần là dữ liệu thuộc kiểu cơ sở/cấu trúc.

Khai báo (định nghĩa) 1 kiểu cấu trúc

```
struct cau_truc_tui  
{  
    //Khai bao cac thanh phan  
};
```

Chú ý: + **struct** là từ khóa (khai báo **kiểu struct**)
+ **cau_truc_tui** tên do người lập trình đặt.
+ dấu ';' là bắt buộc.
+ Thành phần của cấu trúc được khai báo trong khối lệnh.

THỰC HÀNH

Khai báo (định nghĩa) 1 kiểu cấu trúc

```
struct cau_truc_tui
```

```
{
```

```
    //Khai bao cac thanh phan
```

```
};
```

```
=====
```

Ví dụ

```
struct ngaytn
```

```
{
```

```
    unsigned short ngay;
```

```
    unsigned short thang;
```

```
    int nam;
```

```
};
```

THỰC HÀNH

Khai báo (định nghĩa) 1 kiểu cấu trúc

```
struct cau_truc_tui
```

```
{  
    //Khai bao cac thanh phan  
};
```

```
=====
```

Ví dụ

```
struct ngaydl
```

```
{  
    char *thu;  
    struct ngaytn ngay;  
    unsigned short ngaylamviec; //1: ngay lam viec; 0: ngay nghi  
};
```

THỰC HÀNH

Khai báo (định nghĩa) 1 kiểu cấu trúc

```
struct cau_truc_tui
```

```
{
```

```
    //Khai bao cac thanh phan
```

```
};
```

```
=====
```

Ví dụ

```
struct sinvien_vlth
```

```
{
```

```
    char *ht;
```

```
    int mssv;
```

```
    struct ngaytn ngaysinh;
```

```
    int tcdk;
```

```
    int tctl;
```

```
//    cac thuoc tinh khac
```

```
};
```

THỰC HÀNH

Vừa thiết kế kiểu ct vừa sử dụng để khai báo biến

```
struct cau_truc_tui
```

```
{
```

```
    //Khai bao cac thanh phan
```

```
};
```

```
=====
```

Ví dụ

```
struct ngaytn1
```

```
{
```

```
    unsigned short ngay;
```

```
    unsigned short thang;
```

```
    int nam;
```

```
} ngaysinh, ngaydihoc;
```

THỰC HÀNH

Vừa thiết kế kiểu ct để khai báo biến, không định nghĩa kiểu ct mới

```
struct cau_truc_tui
```

```
{
```

```
    //Khai bao cac thanh phan
```

```
};
```

```
=====
```

Ví dụ

```
struct
```

```
{
```

```
    unsigned short ngay;
```

```
    unsigned short thang;
```

```
    int nam;
```

```
} ngaysinh, ngaydihoc;
```


THỰC HÀNH

Khai báo (định nghĩa) 1 kiểu cấu trúc

```
1  struct cau_truc_tui
2  {
3      //Khai bao cac thanh phan
4  };
5  =====
6  #include <stdio.h>
7  void main()
8  {
9      struct ngaytn {
10         unsigned short ngay;
11         unsigned short thang;
12         int nam;
13     };
14     struct ngaytn homnay;
15     printf("\nNhap vao ngay hom nay: ");
16     printf("\nngay: ");scanf("%d",&homnay.ngay);
17     printf("\nthang: ");scanf("%d",&homnay.thang);
18     printf("\nnam: ");scanf("%d",&homnay.nam);
19     printf("\nChuc mot ngay lam viec vui ve\n");
20     printf("\nngay %d/%d/%d",homnay.ngay,homnay.thang,homnay.nam);
21 }
```

THỰC HÀNH

Khai báo (định nghĩa) 1 kiểu cấu trúc

```
1  typedef struct
2  {
3      //Khai bao cac thanh phan
4  } cau_truc_tui;
```

=====

Ví dụ

```
7  typedef struct
8  {
9      unsigned short ngay;
10     unsigned short thang;
11     int nam;
12 } ngaytn2;
```

THỰC HÀNH

Khai báo (định nghĩa) 1 kiểu cấu trúc

```
1 typedef struct
```

```
2 {
```

```
3 //Khai bao cac thanh phan
```

```
4 } cau_truc_tui;
```

```
5 =====
```

Ví dụ

```
7 typedef struct
```

```
8 {
```

```
9 char *thu;
```

```
10 struct ngaytn ngay;
```

```
1 ngaytn2 ngayta;
```

```
2 unsigned short ngaylamviec;//1: ngay lam viec; 0: ngay nghi
```

```
3 } ngaydl1;
```

THỰC HÀNH

Khai báo (định nghĩa) 1 kiểu cấu trúc

```
1  typedef struct
2  {
3      //Khai bao cac thanh phan
4  } cau_truc_tui;
```

```
5  =====
```

Ví dụ

```
7  typedef struct
8  {
9      char *ht;
10     int mssv;
11     struct ngaytn ngaysinh;
12     ngaytn2 ngaydihoc;
13     int tcdk;
14     int tctl;
15     //  cac thuoc tinh khac
16 } sinvien;
```

THỰC HÀNH

Mảng kiểu cấu trúc

```
1 struct cau_truc_tui
2 {
3     //Khai bao cac thanh phan
4 };
5 =====
6 #include <stdio.h>
7 void main()
8 {
9     struct ngaytn {
10         unsigned short ngay;
11         unsigned short thang;
12         int nam;
13     };
14     struct ngaytn homay[2];
15     printf("\nNhap vao ngay hom nay: ");
16     printf("\nngay: ");scanf("%d",&homay[0].ngay);
17     printf("\nthang: ");scanf("%d",&homay[0].thang);
18     printf("\nnam: ");scanf("%d",&homay[0].nam);
19     printf("\nChuc mot ngay lam viec vui ve\n");
20     printf("\nngay %d/%d/%d",homay[0].ngay,homay[0].thang,homay[0].nam);
21 }
```

THỰC HÀNH

Mảng kiểu cấu trúc

```
1  typedef struct
2  {
3      //Khai bao cac thanh phan
4      } cau_truc_tui;
5  =====
6  #include <stdio.h>
7  void main()
8  {
9      typedef struct {
10         unsigned short ngay;
11         unsigned short thang;
12         int nam;
13     } ngaytn;
14     ngaytn homay[2];
15     printf("\nNhap vao ngay hom nay: ");
16     printf("\nngay: ");scanf("%d",&homay[0].ngay);
17     printf("\nthang: ");scanf("%d",&homay[0].thang);
18     printf("\nnam: ");scanf("%d",&homay[0].nam);
19     printf("\nChuc mot ngay lam viec vui ve\n");
20     printf("\nngay %d/%d/%d",homay[0].ngay,homay[0].thang,homay[0].nam);
21 }
```

THỰC HÀNH

Định nghĩa cấu trúc `sinhvien` bao gồm các thông tin:

- 1 +Họ và tên
- 2 +MSSV
- 3 +Thứ tự trong danh sách liệt kê
- 4 +Ngày tháng năm sinh
- 5 +Ngày nhập học
- 6 +Ngày ra trường dự kiến
- 7 +Số TC đăng ký
- 8 +Số TC tích lũy
- 9 +Điểm trung bình toàn phần
- 10 +Điểm trung bình tích lũy.

5 Cài đặt CT quản lý sinh viên: Nhập sinh viên; in thông tin 1 sv ra màn hình (đầy
6 đủ/tùy chọn), in danh sách sv+thông tin (đầy đủ/tùy chọn) ra màn hình; thêm/xóa
7 sinh viên trong danh sách; sắp xếp thứ tự theo tên/họ/mssv/ngày tháng năm
8 sinh/điểm trung bình tích lũy/điểm trung bình toàn phần.

Chương 8

Kiểu tập tin

- Giới thiệu
- Các hàm thao tác với kiểu tập tin
- Minh họa

GIỚI THIỆU

Bao gồm tập tin văn bản thô và tập tin nhị phân.

Dữ liệu “file” được lưu trên bộ nhớ phụ (đĩa) --> tồn tại sau khi CT kết thúc.

Đáp ứng tốt nhu cầu trao đổi, tái sử dụng dữ liệu.

Định vị tập tin: tên tập tin và đường dẫn.

1

2

3

4

5

6

7

8

9

10

1

2

3

4

5

6

7

8

9

20

QUY TRÌNH

Bước 1. Mở tập tin: xác định tên tập tin và đường dẫn,

Bước 2. Sử dụng tập tin khi mở thành công,

+ Đọc dữ liệu từ tập tin

+ Ghi dữ liệu lên tập tin

Bước 3. Đóng tập tin.

CÁC HÀM THAO TÁC

Nguyên mẫu

```
1  
2  
3 FILE *fopen(char *tenfile, char *dactinh);  
4  
5  
6  
7  
8  
9
```

```
10 =====
```

```
1     Mở tập tin kiểu FILE *  
2     tenfile cho biết tên và đường dẫn lưu file.  
3  
4  
5  
6  
7  
8  
9  
20
```

CÁC HÀM THAO TÁC

Nguyên mẫu

```
1  
2  
3 FILE *fopen(char *tenfile, char *dactinh);  
4  
5
```

```
6 int fclose(FILE *tenfile);  
7  
8
```

```
9  
10 =====
```

```
1   Đóng tập tin (FILE *)tenfile  
2  
3  
4  
5  
6  
7  
8  
9  
20
```

CÁC HÀM THAO TÁC

Nguyên mẫu

```
1  
2  
3  
4 int fputc(int ch, FILE *tenfile);  
5  
6  
7  
8  
9
```

```
10 =====
```

```
1 Ghi một ký tự (int)ch lên tập tin (FILE *)tenfile  
2  
3  
4  
5  
6  
7  
8  
9  
20
```

CÁC HÀM THAO TÁC

Nguyên mẫu

1
2
3
4 `int fputc(int ch, FILE *tenfile);`
5

6 `int fputs(const char *ckt, FILE *tenfile);`
7

8
9
10 =====

1 Ghi một chuỗi ký tự (**const char ***)ckt lên tập tin (**FILE ***)tenfile
2
3
4
5
6
7
8
9
20

CÁC HÀM THAO TÁC

Minh họa

```
1  #include <stdio.h>
2
3  void main()
4  {
5      FILE *f1 = fopen("taptin1.txt","w");
6      if(f1) {
7          fputc('\t', f1);
8          fputs("1\tNguyen Chi Linh", f1);
9          fputc('\n', f1);
10         fputc('\t', f1);
1         fputs("2\tVLTH",f1);
2         fclose(f1);
3     }
4 }
```

=====

Đặc tính **(char *)"w"** đề nghị mở tập để ghi dữ liệu "taptin1.txt"

Hoặc mở mới file "taptin1.txt", hoặc xóa dữ liệu cũ nếu file đã tồn tại.

CÁC HÀM THAO TÁC

Nguyên mẫu

```
1  int fgetc(FILE *tenfile);
```

```
6  =====
```

```
7      Đọc một ký tự trong tập tin (FILE *)tenfile
```

```
10 #include <stdio.h>
```

```
1  void main()
```

```
2  {      char kt;
```

```
3          FILE *f1 = fopen("taptin1.txt","r");//"r" mo file de doc du lieu
```

```
4          if (f1) {      kt = fgetc(f1); printf("ki tu vua doc la: %c \n",kt);
```

```
5                          kt = fgetc(f1); printf("ki tu vua doc la: %c \n",kt);
```

```
6                          kt = fgetc(f1); printf("ki tu vua doc la: %c \n",kt);
```

```
7                          fclose(f1);
```

```
8                      }
```

```
9      }
```

```
20 }
```


CÁC HÀM THAO TÁC

Nguyên mẫu

```
1 int fgetc(FILE *tenfile);
```

```
2  
3 char *fgets(char *ckt, int n, FILE *tenfile);
```

```
4  
5  
6 =====
```

7 Đọc một dãy ký tự trong tập tin (FILE *)tenfile

8 Kết thúc khi gặp ký tự xuống dòng hoặc đủ n-1 ký tự.

```
9 #include <stdio.h>
```

```
10  
1 void main()
```

```
2 { char *ckt;
```

```
3 FILE *f1 = fopen("taptin1.txt","r");//"r" mo file de doc du lieu
```

```
4 if (f1) { ckt = fgets(ckt, 10, f1);
```

```
5 printf("Day ki tu vua doc la: %s \n",ckt);
```

```
6  
7 fclose(f1);
```

```
8 }  
9
```

```
20 }
```

CÁC HÀM THAO TÁC

Nguyên mẫu

```
1  int fscanf(FILE *tenfile, char *dinh dang, ...);
```

```
2  
3  
4  
5  
6  =====
```

```
7      Đọc dữ liệu có định dạng (char *)dinh dang  
8      trong tập tin (FILE *)tenfile
```

```
9      #include <stdio.h>
```

```
10  
1  void main()  
2  {  
3      int so;  
4      FILE *f1 = fopen("taptin1.txt","r");//"r" mo file de doc du lieu  
5      if (f1) {  fscanf(f1,"%d",&so);  
6                  printf("Ki tu vua doc la: %d \n",so);  
7  
8                  fclose(f1);  
9      }  
20 }
```

CÁC HÀM THAO TÁC

Nguyên mẫu

```
1 int fscanf(FILE *tenfile, char *dinh dang, ...);
```

```
2  
3 int fprintf(FILE *tenfile, char *dinh dang, ...);
```

```
4  
5  
6 =====
```

```
7 Ghi dữ liệu có định dạng (char *)dinh dang  
8 lên tập tin (FILE *)tenfile  
9  
10
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

```
20
```

CÁC HÀM THAO TÁC

Nguyên mẫu

1 **int fscanf**(**FILE** *tenfile, **char** *dinhtrang, ...);

2
3 **int fprintf**(**FILE** *tenfile, **char** *dinhtrang, ...);

4
5
6 **int fflush**(**FILE** *tenfile);

7
8
9
10 =====

1 Cập nhật dữ liệu lên tập tin (**FILE ***)tenfile

2

3

4

5

6

7

8

9

20

CÁC HÀM THAO TÁC

Nguyên mẫu

1 **int fscanf**(**FILE** *tenfile, **char** *dinhtrang, ...);

2
3 **int fprintf**(**FILE** *tenfile, **char** *dinhtrang, ...);

4
5
6 **int fflush**(**FILE** *tenfile);

7
8 **int flushall**(**FILE** *tenfile);

9
10 =====

1 Cập nhật dữ liệu lên tất cả các tập tin đang mở.
2
3
4
5
6
7
8
9
20

Chương 9

Kỹ thuật duyệt mảng sử dụng con trỏ

- Các phép toán trên địa chỉ, con trỏ
- Duyệt địa chỉ phần tử của mảng
- Hàm trả về địa chỉ

TOÁN TỬ *

Con trỏ vô kiểu

```
1  /*giasu.c*/
2  #include <stdio.h>
3
4  void main()
5  {
6      int i;
7      float b[10];
8      void *z;           //z la bien con tro vo kieu
9
10     z = &i;
11     *(int *)z = 8;      // tuong duong lenh i = 8;
12
13     z = &b[i];
14     *(float *)z = 10.11; // tuong duong lenh b[i] = 10.11
15                          // tuong duong lenh b[8] = 10.11
16     printf("\n b[%d] = %f\n",i,b[i]); // b[8] = 10.110000
17
18 }
19
20
```

CON TRỎ

Thao tác trên địa chỉ bộ nhớ

Cộng trừ với số nguyên: **ptr+n** biểu diễn (giá trị của **ptr**)+n x **sizeof(T)**

Phép trừ hai con trỏ: **ptr1-ptr2** --> (giá trị **ptr1** - giá trị **ptr2**) / **sizeof(T)**
với **ptr1**, **ptr2** là con trỏ cùng kiểu **T** nào đó.

=> **sizeof(void)** không xác định do đó, không thể trừ 2 con trỏ kiểu **void**

Quy ước ký pháp.

+ ***(ctr+sn)** tương đương với **biến** tại địa chỉ **ctr+sn**

+ Sử dụng ký hiệu mảng, **ctr[sn]** truy xuất tương đương **biến** trên.

Trong đó:

+ **biến** được hiểu là có kiểu phù hợp với con trỏ.

+ **sn** là số nguyên, **ctr** là con trỏ hay biến địa chỉ.

+ giả thiết các địa chỉ ở trên đều hợp lệ.

CON TRỎ

Thao tác trên địa chỉ bộ nhớ

```
1  #include <stdio.h>
2  void main()
3  {
4      int *x, y;
5      float z[4], *t;
6      y=5; x = &y; x++
7
8      printf("\n kích thước (con trỏ) x %d",sizeof(x));    //8
9      printf("\n kích thước (địa chỉ) &z[0] %d",sizeof(&z[0]));//8
10     printf("\n kích thước (phần tử) z[0] %d",sizeof(z[0])); //4
1     printf("\n kích thước (tên mảng) z %d",sizeof(z));    //16
2     printf("\n giá trị (con trỏ) x-&y %d",x-&y); //1
3     printf("\n giá trị (con trỏ) x %d",x);    //1781319664
4     printf("\n giá trị (địa chỉ) &y %d",&y);    //1781319660
5     printf("\n giá trị (con trỏ) x+1 %d",x+1); //1781319668
6     printf("\n giá trị (con trỏ) t %d",t);    //1781319904
7     printf("\n giá trị (tên mảng) z %d",z);    //1781319632
8     printf("\n giá trị (địa chỉ) &z[1] %d",&z[1]);//1781319636
9     printf("\n giá trị (địa chỉ) &z[2] %d",&z[2]);//1781319640
20 }
```

DUYỆT PHẦN TỬ MẢNG

Thảo luận: Chỉ số

```
1  #include <stdio.h>
2  int chieudaickt(char *ckt);
3
4  void main()
5  {
6      char *hovaten = "Nguyen Van Abc";
7      int n;
8      n = chieudaickt(hovaten);
9      printf("\n \"%s\" bao gom %d ky tu",hovaten,n);
10 }
11 int chieudaickt(char *ckt) {
12     int cd = 0;
13     while (ckt[cd]!='\0') cd++;
14     return cd;
15 }
16
17
18
19
20
```

DUYỆT PHẦN TỬ MẢNG

Địa chỉ

```
1  #include <stdio.h>
2  int chieudaickt(char *ckt);
3  int chieudaickt2(char *ckt);
4  void main()
5  {
6      char *hovaten = "Nguyen Van Abc";
7      int n;
8      n = chieudaickt(hovaten);
9      printf("\n \"%s\" bao gom %d ky tu",hovaten,n);
10 }
11 int chieudaickt(char *ckt) {
12     int cd = 0;
13     while (ckt[cd]!='\0') cd++;
14     return cd;
15 }
16 int chieudaickt2(char *ckt) {
17     int cd = 0;
18     while (*ckt++) cd++;
19     //toan tu * va ++ co cung uu tien, thu tu: trai <-- phai
20     return cd;
21 }
```

DUYỆT VÀ THAO TÁC TRÊN ĐỊA CHỈ

```
1  #include <stdio.h>
2  int chieudaickt3(char *ckt);
3  int chieudaickt2(char *ckt);
4  void main()
5  {
6      char *hovaten = "Nguyen Van Abc";
7      int n;
8      n = chieudaickt3(hovaten);
9      printf("\n \"%s\" bao gom %d ky tu",hovaten,n);
10 }
11 int chieudaickt3(char *ckt) {
12     char *ptr = ckt;
13     while (*ptr++); //it hon mot nua phep toan
14     return (ptr-ckt-1);//so voi chieudaickt1,2
15 }
16 int chieudaickt2(char *ckt) {
17     int cd = 0;
18     while (*ckt++) cd++;
19     //toan tu * va ++ co cung uu tien, thu tu: trai <-- phai
20     return cd;
21 }
```

HÀM TRẢ VỀ PHẦN TỬ CỦA MẢNG

Thảo luận: Chỉ số

```
1  #include <stdio.h>
2  int chisoNN(int mang[], int nptu);
3  int *controNN(int mang[], int nptu);
4  void main()
5  {
6      int i, n, *ct;
7      int daysn[] = {2, 5, 1, 6, 7};
8      n = sizeof(daysn)/sizeof(daysn[0]);
9      i = chisoNN(daysn,n);
10     ct = controNN(daysn,n);
11     printf("\nso nho nhat %d ?? %d",daysn[i],*ct);
12     printf("\ntai dia chi: 0x%x ?? 0x%x",&daysn[i],ct);
13 }
14 int chisoNN(int src[], int n) {
15     int i = 1; cs = 0;
16     while (i<n){
17         if (src[i] < src[cs]) cs = i;
18         i++;
19     }
20     return cs;
21 }
```

HÀM TRẢ VỀ PHẦN TỬ CỦA MẢNG

Địa chỉ

```

1  #include <stdio.h>
2  int chisoNN(int mang[], int nptu);
3  int *controNN(int mang[], int nptu);
4  void main()
5  {
6      int i, n, *ct;
7      int daysn[] = {2, 5, 1, 6, 7};
8      n = sizeof(daysn)/sizeof(daysn[0]);
9      i = chisoNN(daysn,n);
10     ct = controNN(daysn,n);
11     printf("\nso nho nhat %d ?? %d",daysn[i],*ct);
12     printf("\ntai dia chi: 0x%x ?? 0x%x",&daysn[i],ct);
13 }
14 int *controNN(int src[], int n) {
15     int i = 1; cs = 0;
16     while (i<n){
17         if (src[i] < src[cs]) cs = i;
18         i++;
19     }
20     return &src[cs];

```

HÀM DUYỆT VÀ TRẢ VỀ ĐỊA CHỈ

```
1  #include <stdio.h>
2  int *contross(int [], int , int);
3  int *chepmang(int *, int [], int );
4  void xuatmang(int [], int );
5  void main()
6  {
7      int i, n, gtktra = 7, *ct, nct;
8      int daysn[] = {2, 5, 1, 6, 7, 9, 20};
9      n = sizeof(daysn)/sizeof(daysn[0]);
10     xuatmang(daysn,n);
11     while(ct=contross(daysn,n,gtktra)){
12         nct = n-(ct-daysn);
13         ct = chepmang(ct, ct+1,nct);n--;
14         xuatmang(daysn,n);
15     }
16     void xuatmang(int src[], int n) {
17         int i=0;
18         printf("\ndaysn = {");
19         while (i<n) printf("%d,",src[i++]);
20         printf("nothing};\n");
21     }
```

HÀM DUYỆT VÀ TRẢ VỀ ĐỊA CHỈ

Minh họa: Khử phần tử mảng

```
1  #include <stdio.h>
2  void xuatmang(int [], int);
3  int *chepmang(int *, int [], int);
4  int *contross(int [], int , int);
5  int *khuptumang(int [], int *, int);
6  void main()
7  {
8      int n, gtktra = 5;
9      int daysn[] = {2, 5, 1, 6, 7, 9, 20};
10     n = sizeof(daysn)/sizeof(daysn[0]);
11     xuatmang(daysn,n);
12     khuptumang(daysn, &n, gtktra);
13     xuatmang(daysn,n);
14 }
15 int *khuptumang(int src[], int *n, int gtss) {
16     int *ctr, nct;
17     while (ctr = contross(src,*n,gtss)) {
18         nct = ((*n)--)-(ctr-src);
19         chepmang(ctr, ctr+1, nct);
20     }
21 }
```


XOÁ PHẦN TỬ DÃY KÝ TỰ

Minh họa

```
1  #include <stdio.h>
2  #include <string.h>
3  void khukhoangtrang(char *, char *);
4  void kktctro(char *src);
5
6  void main()
7  {
8      char ckt[] = "Nguy en- Va n -ABC";
9      char ckt2[] = "Nguy en- Va n -ABC";
10     khukhoangtrang(ckt,ckt);
1     puts(ckt);
2     kktctro(ckt2);
3     puts(ckt2);
4 }
5
6 void kktctro(char *src) {
7     char *ctr;
8     while (ctr = strchr(src,' ')) strcpy(ctr,ctr+1);
9 }
20
```

DÂY CON TRONG DÂY ĐÃ CHO

Minh họa: diễn giải hàm

```
1  int *contromcon(int src[], int n, int mss[], int nmc) {
2      int i=0, *a=src, *b, *c;
3      if (mss==NULL) return src;
4      while ((a++)-src<n-nmc) {
5          if(*a==*mss){
6              b = mss; c = a;
7              while (*b++==*c++);
8              if (b>mss+nmc) return a;
9          }
10     }
11     if (a-src>=n-nmc) return NULL;
12 }
```

3
4
5
6
7
8
9
20

DẪY CON TRONG DẪY ĐÃ CHO

Minh họa: Sử dụng hàm

```
1  #include <stdio.h>
2  int *contromcon(int [], int , int [], int );
3  void xuatmang(int [], int);
4
5  void main()
6  {
7      int n, nmc, *ct;
8      int daysn[] = {2, 5, 1, 2, 5, 1, 6, 7, 9, 20};
9      int dayc[] = {5,1,6};
10     int *test = NULL;
11     n = sizeof(daysn)/sizeof(daysn[0]);
12     nmc = sizeof(dayc)/sizeof(dayc[0]);
13     xuatmang(daysn,n);
14     xuatmang(dayc,nmc);
15     ct = contromcon(daysn,n,test,nmc);
16     ct = contromcon(daysn,n,dayc,nmc);
17     if (ct){ printf("\\n ct - daysn = %d",ct-daysn);
18             printf("\\n noi dung tai ct la %d",*ct);
19     }
20 }
```

ĐỀ QUY

Kỹ thuật lặp

```
1  int tong(int a[], int n) {
2      if(n<=0) return 0;
3      return a[n-1] + tong(a,n-1);
4  }
5
6  int giaithua(int n) {
7      if(n<=0) return 1;
8      return n*giaithua(n-1);
9  }
10
1  double tongeuler(int n, int k) {
2      if(n<=0||k<1) return 0;
3      return 1./pow(n,k)+tongeuler(n-1,k);
4  }
5
6
7
8
9
20
```

ĐỆ QUY

Dùng mảng con

```
1 void sapxeptang(int a[], int n) {
2     int i, csn=0, csl=0, *tam;
3     for(i=0;i<n;i++) {
4         if(a[csn]>a[i]) csn = i;
5         if(a[csl]<a[i]) csl = i;
6     }
7     hoanvi(&a[0],&a[csn]);
8     if(!csl) csl = csn;
9     hoanvi(&a[n-1],&a[csl]);
10    if(n>3) {
11        tam = &a[1];
12        sapxeptang(tam,n-2);
13    }
14 }
15
16 void hoanvi(int *a, int *b) {
17     int tam = *a;
18     *a = *b;
19     *b = tam;
20 }
```

Chương 10

Bộ nhớ động và ứng dụng trong danh sách liên kết

- Đặt vấn đề
- Các hàm cấp phát bộ nhớ động
- Khai báo và sử dụng danh sách liên kết đơn
- Minh họa

VÙNG NHỚ

Stack, heap

Stack: Vùng nhớ cần thiết được cấp phát khi gọi hàm.
Tự động giải phóng vùng nhớ sau khi kết thúc gọi hàm.

Heap: Vùng nhớ mang tính toàn cục.
Vùng nhớ được đề nghị cấp phát bởi việc gọi hàm cấp phát.
Vùng nhớ được đề nghị giải phóng bởi việc gọi hàm giải phóng.

CÁC HÀM KHAI THÁC BỘ NHỚ ĐỘNG

Cấp phát

```
1 void *malloc(int dodaibyte);
```

```
10 =====
```

1 Cấp phát ***một vùng bộ nhớ*** từ heap.

2 Số byte là **dodaibyte**.

6 Giá trị của hàm là địa chỉ vùng nhớ được cấp phát

7 Giá trị của hàm là NULL khi không có đủ vùng nhớ để cấp phát.

CÁC HÀM KHAI THÁC BỘ NHỚ ĐỘNG

Cấp phát

```
1 void *malloc(int dodaibyte);
```

```
2  
3 void *calloc(int soptu, int dodaibyteptu);
```

```
4  
5  
6  
7  
8  
9  
10 =====
```

1 Cấp phát **bộ nhớ** từ **heap**. Khởi tạo 0 cho tất cả các byte trong vùng nhớ.
2 Số lượng phần tử được cấp phát là **soptu**
3 Mỗi phần tử có kích thước là **dodaibyteptu**.
4

5
6 Giá trị của hàm là địa chỉ vùng nhớ được cấp phát

7 Giá trị của hàm là NULL khi không có đủ vùng nhớ để cấp phát.
8
9
20

CÁC HÀM KHAI THÁC BỘ NHỚ ĐỘNG

Cấp phát

```
1 void *malloc(int dodaibyte);
```

```
2  
3 void *calloc(int soptu, int dodaibyteptu);
```

```
4  
5 void *realloc(void *ctro, int dodaibyte);
```

```
6  
7  
8  
9  
10 =====
```

1 Cấp phát lại **vùng bộ** đã cấp phát từ **heap**. **ctro** là địa chỉ vùng nhớ cũ.
2 Số byte là **dodaibyte**.

3 Dữ liệu cũ nằm trong phạm vi **dodaibyte** được bảo toàn.

4
5 Giá trị của hàm là địa chỉ vùng nhớ được cấp phát

6 Giá trị của hàm là NULL khi không có đủ vùng nhớ để cấp phát.
7
8
9
20

CÁC HÀM KHAI THÁC BỘ NHỚ ĐỘNG

Cấp phát

```
1 void *malloc(int dodaibyte);
```

```
2  
3 void *calloc(int soptu, int dodaibyteptu);
```

```
4  
5 void *realloc(void *ctro, int dodaibyte);
```

```
6  
7 void free(void *ctro);
```

```
8  
9  
10 =====
```

```
1     Giải phóng vùng nhớ cấp phát có địa chỉ là ctro.
```

```
2     Giá trị ctro phải là địa chỉ vùng nhớ hợp lệ từ heap
```

MINH HỌA

Biết số phần tử trước khi cấp phát

```
1  int *nhapmangp(int *n) {
2      int *tam, i;
3      printf("Nhap so ptu day so nguyen: ");scanf("%d",n);
4      if(*n<=0) return NULL;
5      tam = (int *)calloc(*n,sizeof(int));
6      printf("Nhap ptu day:");
7      if(tam){
8          for(i=0;i<*n;i++) scanf("%d",&tam[i]);
9      }
10     return tam;
11 }
12
13 void xuatmang(int a[], int n) {
14     int i;
15     for(i=0;i<n;i++) printf("%d",a[i]);
16 }
17
18
19
20
```

MINH HỌA

Biết số phần tử trước khi cấp phát

```
1  #include <malloc.h>
2  #include <stdio.h>
3
4  void xuatmang(int [], int);
5  void *nhapmangp(int *);
6
7  void main()
8  {
9      int *a, n;
10     a = nhapmangp(&n);
11     if(a) {
12         xuatmang(a,n);
13         free(a);
14     }
15 }
```

MINH HỌA

Không cho biết số phần tử trước khi cấp phát

```
1  int *themptu(int *day, int *n, int ptu)
2  {  int *tam;
3      tam = (int *)realloc(day,++(*n)*sizeof(int));
4      if(tam) tam[*n-1] = ptu;
5      return tam;
6  }
7  int *nhapmang(int *n)
8  {  int *tam, *day = NULL, ptu;
9      *n = 0;
10     while (scanf("%d",&ptu)) {
1         tam = themptu(day,n,ptu);
2         if (tam) day = tam;
3     }
4     return day;
5 }
6 void xuatmang(int a[], int n)
7 {  int i;
8     for(i=0;i<n;i++) printf("%d",a[i]);
9 }
20
```

MINH HỌA

Không cho biết số phần tử trước khi cấp phát

```
1  #include <malloc.h>
2  #include <stdio.h>
3
4  void xuatmang(int [], int);
5  int *nhapmang(int *);
6  int *themptu(int *, int *, int);
7
8  void main()
9  {
10     int *a, n;
11     printf("Nhap cac ptu day, ket thuc bang Ctrl+x");
12     a = nhapmang(&n);
13     if (a) {
14         xuatmang(a,n);
15         free(a);
16     }
17 }
```

DANH SÁCH LIÊN KẾT ĐƠN

Khai báo

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  //=====
4  typedef struct nut *lket;
5  struct nut {
6      int gtri;
7      lket ktiep;
8  };
9  //=====
```

10

1

2

3

4

5

6

7

8

9

20

DANH SÁCH LIÊN KẾT ĐƠN

Xây dựng

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  //=====
4  typedef struct nut *lket;
5  struct nut {
6      int gtri;
7      lket ktiep;
8  };
9  //=====
10 lket taoNutDon(int k) {
1     lket tam;
2     tam = (lket) malloc(sizeof(struct nut));
3     if (!tam){
4         printf("cap phat nutDon that bai!");exit(0);
5     }
6     tam->gtri = k;
7     tam->ktiep = NULL;
8     return tam;
9 }
20
```

DANH SÁCH LIÊN KẾT ĐƠN

Xây dựng

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  //=====
4  typedef struct nut *lket;
5  struct nut {
6      int gtri;
7      lket ktiep;
8  };
9  //=====
10 lket themNutDau(lket *dau, lket *cuoi, int k) {
1      lket nuttam = taoNutDon(k);
2      if (!*dau)
3          *dau = *cuoi = nuttam;
4      else {
5          nuttam->ktiep = *dau;
6          *dau = nuttam;
7      }
8  }
9
20
```

DANH SÁCH LIÊN KẾT ĐƠN

Xây dựng

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  //=====
4  typedef struct nut *lket;
5  struct nut {
6      int gtri;
7      lket ktiep;
8  };
9  //=====
10 lket themNutCuoi(lket *dau, lket *cuoi, int k) {
1      lket nuttam = taoNutDon(k);
2      if (!*dau)
3          *dau = *cuoi = nuttam;
4      else {
5          (*cuoi)->ktiep = nuttam;
6          *cuoi = nuttam;
7      }
8  }
9
20
```

DANH SÁCH LIÊN KẾT ĐƠN

Luyện tập

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  //=====
4  typedef struct nut *lket;
5  struct nut {
6      int gtri;
7      lket ktiep;
8  };
9  //=====
10 void chenNutSau(lket nutmoc, int k);
1
2 void chenNutTruocp(lket dau, lket nutmoc, int k);
3
4 void xoaNutGiua(lket dau, lket bo);
5
6 void xuatDanhSach(lket dau);
7
8
9
20
```