**实验四：选择结构程序设计**

1. **目的要求**
2. 掌握C语言的关系运算符和关系选择表达式。
3. 掌握逻辑运算符和逻辑表达式，学会表示逻辑值的方法。
4. 熟练掌握条件语句和多分支语句，学习选择结构程序设计的方法及应用。
5. **上机实验**

上机题1：编写程序。

上机题2：输入并运行一下程序，分析程序的运行结果。

# include <iostream.h>

void main()

{

int a, b;

cin>>a>>b;

if (a > b)

{

cout<<a<<endl;

}

else

{

cout<<b<<endl;

}

}

修改后：

# include <iostream.h>

void main()

{

int a, b, c;

int max;

cin>>a>>b>>c;

if (a > b)

{

if (a > c)

{

max = a;

}

else

{

max = c;

}

}

else

{

if (b > c)

{

max = b;

}

else

{

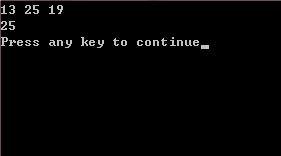
max = c;

}

}

cout<<max<<endl;

}



上机题3：程序实现求分段函数的值，完成程序并填空。

/\*# include <iostream.h>

# include <iomanip.h>

void main()

{

float x, y;

cin>>x;

if (x <= -10)

{

y = 2 \* x;

}

else

{

if (x <= 0)

{

y = 2 + x;

}

else

{

if (x <= 10)

{

y = x - 2;

}

else

{

y = x / 10;

}

}

}

cout<<setiosflags(ios::showpoint)<<"x = "<<x<<"y = "<<y<<endl;

}

# include <iostream.h>

void main()

{

float x;

char y;

cin>>x;

if (x < 60)

{

y = 'E';

}

else

{

if (x < 70)

{

y = 'D';

}

else

{

if (x < 80)

{

y = 'C';

}

else

{

if (x < 90)

{

y = 'B';

}

else

{

y = 'A';

}

}

}

}

cout<<y<<endl;

}

\*/

# include <iostream.h>

void main()

{

float x;

int i, y;

cin>>x;

y = x / 10;

if (y <= 5)

{

i = 5;

}

switch (i)

{

case 5: cout<<"E"<<endl;

break;

case 6:cout<<"D"<<endl;

break;

case 7:cout<<"C"<<endl;

break;

case 8:cout<<"B"<<endl;

break;

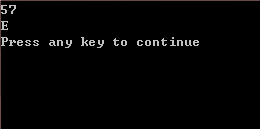
case 9:cout<<"A"<<endl;

break;

default:cout<<"A"<<endl;

}

}



上机题4：输入并运行以下程序，分析程序运行结果。

# include <iostream.h>

void main()

{

int a, b, c;

cin>>a>>b>>c;

switch (a)

{

case 1:

case 2:

case 3: cout<<b<<"+"<<c<<"="<<b + c<<endl;

case 4: cout<<b<<"-"<<c<<"="<<b - c<<endl;

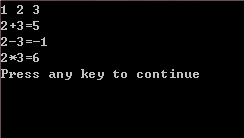
case 5: cout<<b<<"\*"<<c<<"="<<b \* c<<endl;

case 6: cout<<b<<"/"<<c<<"="<<b / c<<endl;

default: cout<<"a的值不正确!"<<endl;

}

}



上机题5：程序实现求分段函数的值，完成程序并填空。

# include <iostream.h>

# include <iomanip.h>

void main()

{

float x, y;

int grade = 0, flag = 0;

cin>>x;

if (x > -5 && x < 0)

{

grade = 1;

}

if (x == 0)

{

grade = 2;

}

if (x > 0 && x < 8)

{

grade = 3;

}

switch (grade)

{

case 1: y = x - 1;

break;

case 2: y = x;

break;

case 3: y = x + 1;

break;

default: cout<<"x的值超出范围"<<endl;

flag = 1;

}

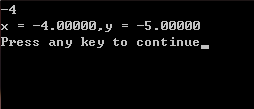
if (! flag)

{

cout<<setiosflags(ios::showpoint)<<"x = "<<x<<","<<"y = "<<y<<endl;

}

}



1. **小结**

C语言的运算符主要用于构成表达式，同一个符号在不同的表达式中，其作用并不一致。下面按计算的优先顺序，分别说明不同作用的表达式。需要特别指出，在C语言标准中，并没有结合性的说法。

相同优先级运算符，从左至右依次运算。注意后缀运算优先级高于前缀。因此++i++应解释为++(i++)。

而与或非的运算优先级都不一样，因此a && b || b && c解释为(a && b) || (b && c)

合理使用优先级可以极大简化表达式。