C语言时一门结构化的程序设计语言

1.顺序结构

2.选择结构: if switch

3.循环结构: whilefordo while

分支语句

if else if else

如果条件成立，要执行多个语句，应使用一对代码块 { }

#include <stdio.h>

int main()

{

    int age = 11;

    if(age<18) {

        printf("未成年\n");

        printf("不能谈恋爱\n");

    }

    else if  (age>=18 && age<28)

        printf("青年\n");

    else

        printf("壮年\n");

    return 0;

}

悬空else

else 只与距离最近的未匹配的 if 进行匹配

#include <stdio.h>

int main()

{

    int a = 0;

    int b = 2;

    if (a == 1)

        if(b == 2)

            printf("hehe\n");

    else //此else与if(b==2)进行匹配

        printf("haha\n");

    return 0;

}

//此时没有输出结果

if 书写格式的对比

if (condition) {

return x;

}

return y;

//此时满足时返回 x ,不满足条件时返回 y ;

//规范书写

if (condition) {

return x;

}

else{

return y;

}

//

int mun = 1;

if (num == 5){

printf("haha\n");

}

//规范书写

int mun = 1;

if (5 == num){

printf("haha\n");

}

分支语句

while 找出100以内奇数

#include <stdio.h>

int main()

{

    int i = 0;

    while (i<=100)

    {

        if (i%2 == 1){

            printf("%d\n",i);

        }

        i++;

    }

    return 0;

}

switch - 常用于多分支语句

switch(整形表达式) //必须是整形表达式

{

case 整形常量表达式://必须是整形常量表达式

printf()

break;

default:

break;

}

switch 允许嵌套

#include <stdio.h>

int main()

{

    int day = 0;

    scanf("%d",&day);

    switch (day)

    {

    case 1:

        printf("星期1\n");

        break;

    case 2:

        printf("星期2\n");

        break;

    case 3:

        printf("星期3\n");

        break;

    }

default:

break;

    return 0;

}

#include <stdio.h>

int main()

{

    int day = 0;

    scanf("%d",&day);

    switch (day)

    {

    case 1:

    case 2:

    case 3:

        printf("工作日\n");

        break;

case 6:

printf("休息日\n");

break;

    }

default:

break;

    return 0;

}

while

continue - 当运行到continue时，终止此次循环，不再执行下面语句，直接返回到最一开始

#include <stdio.h>

int main()

{

    int i = 1;

    while (i<=10)

    {

        if (i == 5)

            continue;   //当运行到continue时，不再执行下面语句，直接返回到最一开始-while

            //break;   当i=5时，结束次循环

        printf("%d",i);

        i++;

    }

    return 0;

}

getchar() 输入字符函数

putchar() 输出字符函数

#include <stdio.h>

int main()

{

    int ch = getchar();

    putchar(ch);

    printf("%c\n",ch);

    return 0;

}