C范例代码训练

邢文鹏制作

张继瑀修订

第一章c语言程序基本结构

1函数的基本结构

第一个c语言程序

1、用c语言实现”helloworld!”字符串的打印

```
#include <stdio.h>

int main(void)
{
    printf("Hello World!\n");
    return 0;
}
```

2、函数返回值—return 语句

```
#include <stdio.h>

int foo(void)
{
return 5;
}

int main(int argc, char *argv[])
{
    int a = foo();
    return a;
}
```

3、全局变量及同名局部变量

```c
#include <stdio.h>

int b;
int c;
int d;
int main(int argc, char *argv[])
{
     int a;
     int c= 5;
     int d = 6;
     return 0;
}
```

自定义函数

1、函数的定义、声明

```c
#include <stdio.h>

int goo(int d, int e, int f)
{
     int g = d + e;
     return f;
}

int swap(int a, int b);

int foo()
{
     return 0;
}

int main(int argc, char *argv[])
{

     int x, y;
     x = -1; y = 2;

     printf("%d,%d\n", x, y);
     swap(x,y);
     printf("%d,%d\n", x, y);
     printf("%d\n", goo(2, 3, 4));

     foo(1, 2, 34, 5);

     return 0;
}

int swap(int a, int b)
```

```
{
	int tmp = a;
	a = b;
	b = tmp;

	return 0;
}
```

2、自定义函数应用

```
/*
- main.c - demo helloworld application
*/

#include <stdio.h>

int add(int a, int b)
{
	return a + b;
}

int sub(int a, int b)
{
	return a - b;
}

int main(int argc, char * argv[])
{
	int ret;
	printf("hello, Cruel World! \n");
	ret = add(100, 200);
	printf("add = %d \n", ret);
	ret = sub(100, 200);
	printf("sub = %d \n", ret);

	return 0;
}
```

2、变量、常量和表达式

1、数值的格式化打印

```
#include <stdio.h>

int foo(int n)
{
	printf("%d\n", n);
```

```
        return 0;
}

int main(int argc, char *argv[])
{
        printf("%d\n", 100);
        printf("%f\n", 100.0);
        foo(100.0);

        return 0;
}
```

2、sizeof 的使用及数值的格式化打印

```
#include <stdio.h>

int main(int argc, char *argv[])
{

        char *a;
        printf("%d\n", sizeof a); printf("%d\n", sizeof *a);

        int b, c;
        b = 1;
        c = 1;

        printf("%d\n", sizeof(++b));
        printf("%d\n", 4);
        printf("%d\n", sizeof(int));

        /* short int long longlong */ /* u, d, x, o */

        printf("%u\n", 0xFFFFFFFF);
        printf("%u\n", -1);
        printf("%d\n", 0xFFFFFFFF);
        printf("%d\n", -1);
        printf("%x\n", 0xFFFFFFFF);
        printf("%x\n", -1);
        printf("%o\n", 0xFFFFFFFF);
        printf("%o\n", -1);
        return 0;

}
```

3、浮点数的打印

```
#include <stdio.h>

int main(int argc, char *argv[])
{
```

```
	double d1 = 12.0, d2 = 12.0;

	printf("%f\n", (1.0E20 - 1.0E20) + 3.14);
	printf("%f\n", 1.0E20 - (1.0E20 + 3.14));

	if (d1 == d2)
	{
		printf("Hello\n");
	}
	return 0;
}
```

4、变量及转义字符

```
#include <stdio.h>

int main(int argc, char *argv[])
{

	printf("\\t\taaa\bbbb\n");
	printf("int:%d,ox%x,double:%f\n", 7, 100, 27.0);
	printf("17.2365: %f\n", 17.2365);
	printf("%%d,',\"\n");

	printf("sizeof(int) = %d\n", sizeof(int));
	printf("sizeof(short) = %d\n", sizeof(short));
	printf("sizeof(long) = %d\n", sizeof(long));
	printf("sizeof(char) = %d\n", sizeof(char));
	printf("sizeof(float) = %d\n", sizeof(float));
	printf("sizeof(double) = %d\n", sizeof(double));
	return 0;

}
```

5、全局变量

```
#include <stdio.h>

int a;

int main(int argc, char *argv[])
{

	a += 1;
	printf("%p:%d\n", &a, a);

	while (1)
	{
		a++;
	}
```

```
    return 0;
}
```

6、变量打印

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    double x;
    x = 25 / 2;

    printf("int 25/2 = %d\n", 25 / 2);
    printf("double 25/2 = %f\n", 25.0 / 2);
    printf("double x = %f\n", x);

    return 0;
}
```

7、字符串打印

```
#include <stdio.h>
#include <string.h>
#define N 30

int main(int argc, char *argv[])
{
    char c = 'A';
    char a[N] = "wang_lei.1982@abc.com";
    int i;

    printf("c = %c\n", c); if ((c >= 'A') && (c <= 'Z'))
    printf("A to lower %c\n", c + 32);
    printf("49:%c, %d\n", 49, 49);
    printf("len of char array = %d\n", sizeof(a) / sizeof(char));
    printf("strlen(a) = %d\n", strlen(a));
    printf("Email address:");

    for (i = 0; i < strlen(a); i++)
    {
        printf("%c", a[i]);
    }
    printf("\n");

    return 0;
}
```

8、字符的大小写转换

```c
#include <stdio.h>

int main(int argc, char *argv[])
{

printf("%c\n", ('A' +32));

return 0;
}
```

9、条件运算符的应用

```c
#include <stdio.h>

int main(int argc, char *argv[])
{

     int a[100];
     int i;

     for (i = 0; i < 100; i++)
     {
           a[i] = 100 + i;
           printf("%d%c", a[i], (i % 5 != 4) ? '\t' : '\n');
     }
     return 0;
}
```

10、类型转换–typeconv.c

```c
#include <stdio.h>

int main(void)
{
     int i = 0x1ffff;
     printf("%hx\n", (unsigned short)i);
     return 0;
}
```

11、类型提升–intpromo.c

```c
#include <stdio.h>

void foo(double d)
{
```

```
    printf("%f\n", d);

}

int main(void)
{

    void foo(double);
    char c = 60; foo(c);
    return 0;

}
```

12、打印变量的地址

```
#include <stdio.h>
#define MAX 5
#define MIN 0

int globle_var = 0x1234;
const int const_num = 0x9999;
int add(int, int);
int a_out = 10;
int b_out = 20;

int main(int argc, char * argv[])
{

    int a = 100; int b = 200;
    int * p = 0x804972c;

    printf("<main.c> address a = 0x%x \n", &a);
    printf("<main.c> address b = 0x%x \n", &b);
    printf("<main.c> aaddress a_out = 0x%x \n", &a_out);
    printf("<main.c> aaddress b_out = 0x%x \n", &b_out);

    // return 0;
    add(a, b); //return 0;
    *p = 123;
    add(a, b);

    //printf( "sta_a = %d \n", *p);

    return 0;

}
int adda = 1;
int addb = 2;

int add(int a, int b)

{
```

```
	int aa = 1; int bb = 1;

	static int sta_a = 1;

	printf("<add.c> address static a = 0x%x \n", &sta_a);
	printf("<add.c> static a = %d \n", sta_a);

	printf("<add.c> address a = 0x%x \n", &a);
	printf("<add.c> address b = 0x%x \n", &b);

	printf("<add.c> address aa = 0x%x \n", &aa);
	printf("<add.c> address bb = 0x%x \n", &bb);

	printf("<add.c> address adda = 0x%x \n", &adda);
	printf("<add.c> address addb = 0x%x \n", &addb);

	return a + b;

}
```

13、栈上变量分析

```
#include <stdio.h>

int * addr = (int *)0;
int x = 0;
int test(int para)
{
	int i[1000];
	static int j;
	printf("<func> &para = 0x%x \n", &para);

#if 0

	printf("<func> para = 0x%x \n", para);
	printf("<func> &i = 0x%x \n", &i);
	printf("<func> i = 0x%x \n", i);

	printf("<func> &j = 0x%x \n", &j);
	printf("<func> j = 0x%x \n", j);
#endif

	addr = (int *)&i;
	printf("<func> --- out --- \n");

	x++;
	if (5 == x)
	{
		return 0;
	}
	test(5);
```

```
}

int test2(int para, int p2, int p3, int p4, int p5)
{

	int k1;
	int k2;
	static int jj;

	printf("<func2> &para = 0x%x \n", &para);
	printf("<func2> para = 0x%x \n", para);
	printf("<func2> &p2 = 0x%x \n", &p2);
	printf("<func2> &p3 = 0x%x \n", &p3);
	printf("<func2> &p4 = 0x%x \n", &p4);
	printf("<func2> &p5 = 0x%x \n", &p5);
	printf("<func2> p2 = 0x%x \n", p2);
	printf("<func2> p3 = 0x%x \n", p3);
	printf("<func2> p4 = 0x%x \n", p4);
	printf("<func2> p5 = 0x%x \n", p5);

	printf("<func2> &k1 = 0x%x \n", &k1);
	printf("<func2> k1 = 0x%x \n", k1);

	printf("<func2> &k2 = 0x%x \n", &k2);
	printf("<func2> k2 = 0x%x \n", k2);

	printf("<func2> &jj = 0x%x \n", &jj);
	printf("<func2> jj = 0x%x \n", jj);

	//addr = (int *) &i;

	printf("<func2> --- out --- \n");
	return 0;

}

int main(int argc, char * argv[])
{
	test(5);
	*(int *)(addr) = 0x3344;
	*(int *)(addr - 1) = 0x5566;
	test2(5, 6, 7, 8, 9);
	return 0;

}
```

3、分支结构

1、if/else 分支结构

```c
#include <stdio.h>

int main(int argc, char *argv[])
{

	char c;
	c = 128;
	if (c == 128)
	{
		printf("Hello\n");
	}
	else
	{
		printf("World\n");
	}
	printf("%d\n", c);

	return 0;
}
```

2、计算数组中的奇偶数的个数

```c
#include <stdio.h>

#include <stdlib.h>
#include <math.h>
#include <time.h>

int main(int argc, char *argv[])
{
	int a[50];
	int i;
	int oddsum = 0, evensum = 0;

	srand((unsigned)time(NULL));
	for (i = 0; i < 50; i++)
	{
		a[i] = rand() % 30;
		printf("%d%c", a[i], (i % 5 != 4) ? '\t' : '\n');

	}

	for (i = 0; i < 50; i++)
	{
		if (a[i] % 2)
		{
			evensum++;
		}
		else
		{
			oddsum++;
```

```
            }
      }

      printf("In array a, %d numbers are even, %d numbers are odd!\n", e
vensum, oddsum);

      if (15 % 4 == 15 / 4)
      {
            printf("equal\n");
      }
      else
      {
            printf("not equal\n");
      }
      return 0;
}
```

3、判断闰年

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

int is_leap_year(int year)
{
      if ((year % 4 == 0) && (year % 100 != 0))
      {
            return 1;
      }
      else if (year % 400 == 0)
      {
            return 1;
      }
      else
      {
            return 0;
      }

}

int main(int argc, char *argv[])
{

      int year;
      if (argc != 2)
      {
            printf("Usage:invariable argc, please input year!\n"); retur
n 0;
      }

      year = atoi(argv[1]);
      if (is_leap_year(year))
      {
```

```
            printf("%d is leap year!\n", year);
      }
      else
      {
            printf("%d is not leap year!\n", year);
      }
      printf("%d is%s leap year!\n", year, is_leap_year(year) ? " " : "
not");

      return 0;
}
```

4、编写一个函数 double myround(double x)，输入一个小数，将它四舍五入

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

double myround(double x)
{

      double y;

      if (x - floor(x) < 0.5)
      {
            printf("%f floor is %f\n", x, floor(x)); y = floor(x);
      }
      else
      {
            printf("%f ceil is %f\n", x, ceil(x)); y = ceil(x);
      }
      return y;
}

int main(int argc, char *argv[])
{
      double x;
      if (2 != argc)
      {
            printf("Usage:argc,please input double!\n"); return 0;
      }

      x = atof(argv[1]);

      printf("After round, %f is %f\n", x, myround(x));

      return 0;
}
```

5、求数值的绝对值

```
#include <stdio.h>

int myabs(int x)
{
	if (x >= 0)
	{
		return x;
	}
	else
	{
		return -x;
	}
}

int main(int argc, char *argv[])
{
	int a = 10;
	if (-5)
	{
		printf("True\n");
	}
	else
	{
		printf("False\n");
	}

	printf("-5 abs is %d\n", myabs(-5));
	return 0;

}
```

6、编程判断 3 人中谁的年龄最大，并打印最大者的年龄

```
#include <stdio.h>

int age_compare(int age1, int age2)
{

	if (age1 > age2)
	{
		return age1;
	}
	else
	{
		return age2;
	}
}

int main(int argc, char *argv[])
```

```
{
	int Tom_age, Bob_age, Mike_age;

	printf("Please input Tom Bob Mike age:");
	scanf("%d, %d, %d", &Tom_age, &Bob_age, &Mike_age);

	if (age_compare(Tom_age, Bob_age) == Tom_age)
	{
		printf("%d is the oldest!\n", age_compare(Tom_age, Mike_age)
);
	}
	else if (age_compare(Tom_age, Bob_age) == Bob_age)
	{
		printf("%d is the oldest!\n", age_compare(Bob_age, Mike_age)
);
	}
	return 0;
}
```

7、输入 3 个数，判断这 3 个数是否能构成三角形

```
#include <stdio.h>

int is_triangle(int a, int b, int c)
{
	if (a <= 0 || b <= 0 || c <= 0)
	{
		printf("Usage: a: %d, b: %d, c: %d\n", a, b, c); return -1;
	}
	if (a + b > c && a + c > b && b + c > a)
	{
		return 1;
	}
	else
	{
		return 0;
	}
}

int main(int argc, char *argv[])
{
	int a, b, c;
	int result;
	printf("Please input 3 number as triangle:");
	scanf("%d,%d,%d", &a, &b, &c);

	result = is_triangle(a, b, c);
	if (result == 1)
	{
		printf("a = %d, b = %d, c = %d can found a triangle!\n", a,
b, c);
	}
```

```
    else if (result == -1)
    {
        printf("Invariable a = %d, b = %d, c = %d!\n", a, b, c);
    }
    else
    {
        printf("a = %d, b = %d, c = %d can not found a triangle!\n",
 a, b, c);
    }
    return 0;
}
```

8、switch 分支结构实现一个函数，它能根据参数进行加法、减法、乘法、除法、取模运算

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

int main(int argc, char *argv[])

{

    int a, b; int i; char c;

    a = atoi(argv[1]); b = atoi(argv[2]);

    printf("argc = %d\n", argc);
    for(i = 0; i < argc; i++)
    {
        printf("%s\n", argv[i]);
    }

    c = getchar();
    printf("getchar c = %c\n", c);
    printf("%d %d %c\n", a, b, *argv[3]);
    switch(*argv[3])
    {

        case '*':printf("%d * %d = %d\n", a, b, a*b);
        {
            break;
        }
        case '-':printf("%d - %d = %d\n", a, b, a-b);
        {
            break;
        }
    }

return 0;

}
```

4、循环结构

1、将 1-200 间不能被 3 整除的数输出

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
	int i;

	printf("1-200 / 3 !=0:\n");
	for (i = 1; i <= 200; i++)
	{
		if (i % 3)
		{
			printf("%d ", i);
		}
	}
	printf("\n");
	return 0;
}
```

2、数学表达式 25！结果末尾有多少个 0

```c
#include <stdio.h>

int main(int argc, char *argv[])
{

	int i;
	int sum = 0;
	for (i = 5; 25 / i > 0; i *= 5)
	{
		sum += 25 / i;
	}
	printf("%d '0' in 25!\n", sum);

	return 0;

}
```

3、求表达式值： sum = 1 + 3 + 5 + … + 99

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
	int i;
```

```
    int sum = 0;
    for (i = 1; i <= 99; i += 2)
    {
        sum += i;
    }
    printf("1+3+5+...+99 = %d\n", sum);
    return 0;
}
```

4、有 1、2、3、4 个数字，能组成多少个互不相同且无重复数字的三位数？都是多少？

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    int i, j, k;
    int count = 0;
    for (i = 1; i <= 4; i++)
    {
        for (j = 1; j <= 4; j++)
        {
            for (k = 1; k <= 4; k++) if (i != j && i != k && j != k)
            {
                count++;
                printf("%d%d%d ", i, j, k);
            }
        }
    }
    printf("\ncount = %d\n", count);

    return 0;
}
```

5、打印出右半个菱形

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

int main(int argc, char *argv[])
{
    int i, j, k;
    int n;

    while (scanf("%d", &n))
    {
        if (n & 1)
        {
            break;
        }
```

```
    }
    k = n / 2;

    printf("k = %d\n", k);
    for (i = -k; i <= k; i++)
    {
        if (i < 0)
        {
            for (j = 0; j < i + k + 1; j++)
            {
                printf("*");
            }
        }
        else
        {
            for (j = 0; j < -i + k + 1; j++)
            {
                printf("*");
            }
        }
        printf("\n");
    }
    return 0;
}
```

6、统计一下某字符串中某指定字符出现的次数

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    int result = 0;
    int i;
    char a[20] = "Hello,World!";

    for (i = 0; i < 20; i++)
    {
        if ('o' == a[i])
        {
            result++;
        }
    }
    printf("%d 'o' in \"Hello,World!\"\n", result);

    return 0;
}
```

7、替换字符串中的指定字符为大写，并打印结果

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

```c
#define N 20

int main(int argc, char *argv[])
{
     char a[N] = "I like this game";
     int num = 0;
     int i;
     for (i = 0; i < N; i++)
     {
          if (a[i] >= 'a' && a[i] <= 'z' && a[i] == 'e')
          {
               a[i] = 'e' - 32; num++;
          }
     }
     if (num)
     {
          printf("After converse,the new string: %s\n", a);
     }
     else
     {
          printf("Has not character 'e' need to converse!\n");
     }
     return 0;
}
```

8、寻找二维数组中最小值及其坐标位置 i、j，并打印输出

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
    int a[3][4] = { { 7, 20, 8, 5 },
                    { 55, -2, 64, 72 },
                    { 108, 99, 23, -35 } };
    int max = a[0][0], min = a[0][0];

    int max_i, max_j;
    int min_i, min_j;
    int i, j;

    for (i = 0; i < 3; i++) for (j = 0; j < 4; j++)
    {
          if (a[i][j] > max)
          {
               max = a[i][j]; max_i = i; max_j = j;
          }
          if (a[i][j] < min)
          {
               min = a[i][j]; min_i = i; min_j = j;
          }
    }
    printf("Max is a[%d][%d]=%d, ,Min is a[%d][%d] = %d in array a!\n",
```

```
 max_i, max_j, max, min_i, min_j, min);

    return 0;
}
```

9、输入一个整数数组 a[] = {1, 2, 3, 5, 6, 7, 8, 9, 0, 11, 22, 33, 44, 55,66},调整数组中数字的顺序，使得所有奇数位于数组的前半部分，所有偶数位于数组的后半部分

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    int a[] = { 1, 2, 3, 5, 6, 7, 8, 9, 0, 11, 22, 33, 44, 55, 66 };
    int len = sizeof(a) / sizeof(int);
    int i = 0, j = len - 1;
    int tmp;
    printf("Before modified:\n");
    for (i = 0; i < len; i++)
    {
          printf("a[%d] = %d\n", i, a[i]);
    }
    for (i = 0, j = len - 1; i < j; i++)
    {
        if (a[i] % 2 == 0)
        {
            while (a[j] % 2 == 0)
            {
                j--;
            }
            tmp = a[i];
            a[i] = a[j];
            a[j] = tmp;
            j--;
        }
    }
    printf("After modified:\n");
    for (i = 0; i < len; i++)
    {
         printf("a[%d] = %d\n", i, a[i]);
    }
    printf("sizeof(a) = %d, len = %d\n", sizeof(a), len)
    return 0;
}
```

10、打印菱形–diamond.c

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

```c
void diamond(int n, char c)
{
    int i, j;
    if (n <= 0 || !(n % 2))
    {
        return;
    }
    for (i = 1; i <= n / 2 + 1; i++) {

    for (j = 1; j <= n / 2 - i + 1; j++)
    {
         printf("\t");
    }
    for (j = 1; j <= 2 * i - 1; j++)
    {
         printf("%c\t", c);
    }
         printf("\n");
    }

    for (i = 1; i <= n / 2; i++)
    {
        for (j = 1; j <= i; j++)
        {
            printf("\t");
        }
        for (j = 1; j <= n - 2 * i; j++)
        {
            printf("%c\t", c);
        }
        printf("\n");
    }
}

int main(void)
{
    diamond(11, '+');

    return 0;
}
```

break、continue 与 goto

1、break 和 goto 的用法–hello.h hello.c

```c
/*
main.c - demo helloworld application
*/

#include <stdio.h>
```

```c
int main(int argc, char * argv[])
{

    int i, j;
    printf("hello, Cruel World! \n");

    for (i = 0; i < 4; i++)
    {
        for (j = 0; j < 10; j++)
        {
            printf("(i,j) = (%d,%d) \n", i, j);
            if (2 == j)
            {//break;
                goto out;
            }
        }
    }
out:
    return 0;

}
```

2、求 1-100 的素数

```c
#include <stdio.h>

int is_prime(int n)
{

    int i;
    for (i = 2; i < n; i++)
    {
        if (n % i == 0)
        {
            break;
        }
    }
    if (i == n)
    {
         return 1;
    }
    else
    {
         return 0;
    }
}

int main(void)
{

    int i;
    for (i = 1; i <= 100; i++)
```

```
    {
        if (!is_prime(i))
        {
        continue;
        }
        printf("%d\n", i);
    }

    return 0;
}
```

5、递归函数

1、使用递归函数求和

```
#include <stdio.h>

int sum(int n)
{

    if (0 == n)
    {
        return 0;
    }
    else
    {
        return n + sum(n - 1);
    }
}

int main(int argc, char *argv[])
{
     int n;
     printf("Please input a number:");
     scanf("%d", &n);
     printf("1+2+...+%d = %d\n", n, sum(n));
     return 0;
}
```

2、递归实现 n!

```
#include <stdio.h>

long factorial(int n)
{

    if (0 == n)
    {
```

```
        return 1L;
    }
    else
    {
        return n*factorial(n - 1);
    }
}

int main(int argc, char *argv[])
{
    int n;
    printf("Please input a number < 15:");
    scanf("%d", &n);
    printf("%d! = %ld\n", n, factorial(n));
    return 0;

}
```

3、递归实现求 Fibonacci 数列的第 n 项

```
#include <stdio.h>

#include <assert.h>

int fibonacci(int n)
{

    assert(n >= 0);
    if (n == 0 || n == 1)
    {
        return 1;
    }
    else
    {
        printf("fibonacci(%d) = %d, fibonacci(%d) = %d\n", n - 1, fibon
acci(n - 1), n - 2,fibonacci(n - 2));

        return fibonacci(n - 1) + fibonacci(n - 2);
    }

}

int main(int argc, char *argv[])
{
    int n;

    printf("Please input a number:");
    scanf("%d", &n);
    printf("In main func,fibonacci(%d) = %d\n", n, fibonacci(n));
    return 0;

}
```