**C语言代码合集**

**1.三角形判别**

#include<stdio.h>

#include<math.h>

int main()

{

double a,b, c,t;

printf("Enter three edges of a triangle");

scanf("%lf,%lf,%lf", &a, &b, &c);

if (a <= 0 || b <= 0 || c <= 0)

{

printf("error input!\n");

}

else

{

if (a > b)

{

t = a;

a = b;

b = t;

}

if (a > c)

{

t = a;

a = c;

c = t;

}

if (b > c)

{

t = b;

b = c;

c = t;

}

if (a + b < c)

{

printf("不能构成三角形\n");

}

else

{

if (a \* a + b \* b - c \* c<1E-2)

{

if (a == b)

{

printf("是等腰直角三角形\n");

}

else

{

printf("是直角三角形\n");

}

}

else

{

if (a == b || b == c)

{

if (a == c)

{

printf("是等边三角形\n");

}

else

{

printf("是等腰三角形\n");

}

}

else

{

printf("是一般三角形\n");

}

}

}

}

return 0;

}

**2.用Switch计算天数**

#include<stdio.h>

int main()

{

int year, mouth, daysum;

printf("Enter the year and the mouth:");

scanf("%d%d", &year, &mouth);

switch (mouth)

{

case 1:

case3:

case 5:

case 7:

case 8:

case 10:

case 12:

daysum = 31;

break;

case 4:

case 6:

case 9:

case 11:

daysum = 30;

break;

case 2:

if ((year % 400 == 0) || (year % 4 == 0 && year % 100 != 0))

{

daysum = 29;

}

else

{

daysum = 28;

}

}

printf("%d.%d has %d days.\n", year, mouth, daysum);

return 0;

}

**3.数列前1000项求和**

#include<stdio.h>

int main()

{

int i, sign;

double item, sum;

sum = 0;

sign = 1;

for (i = 1; i <= 1000; i++)

{

item = sign / (2.0 \* i - 1);

sum += item;

sign = -sign;

}

printf("sum=%f\n", sum);

return 0;

}

**4.** **质数判断**

#include<stdio.h>

#include<math.h>

int main()

{

int n, i, k;

do

{

printf("Enter a positive integer:");

scanf("%d", &n);

} while (n <= 0);

if (n == 1)

{

printf("%d is not a prime.\n", n);

}

else

{

k = (int)sqrt(n);

for (i = 2; i <= n; i++)

{

if (n % i == 0)

{

break;

}

}

if (i > k)

{

printf("%d is a prime.\n", n);

}

else

{

printf("%d is not a prime.\n", n);

}

}

return 0;

}

**5.** **求s P72.6**

#include<stdio.h>

#include<math.h>

int main()

{

double s, item, m;

int sign, n;

scanf("%lf", m);

s = 0;

n = 1;

item = 1;

sign = 1;

do

{

item = item \* m / n;

s += sign \* item;

sign = -sign;

n++;

} while (fabs(item) >= 1E-4);

printf("%.2f\n", s);

return 0;

}

**6.打印图形 P72.7**

#include<stdio.h>

int main()

{

int i, j;

for (i = 1; i <= 5; i++)

{

for (j = 1; j <= i - 1; j++)

{

printf(" ");

}

for (j = 1; j <= 11 - 2 \* i; j++)

{

printf("\*");

}

printf("\n");

}

return 0;

}

#include<stdio.h>

#include<math.h>

int main()

{

int i, j;

for (i = 1; i <= 5; i++)

{

if (i <= 3)

{

for (j = 1; j <= 3 - i; j++)

{

printf(" ");

}

}

for (j = 1; j <= 3 - fabs(3 - i); j++)

{

printf("\*");

}

printf("\n");

}

return 0;

}

**7.以每行8个输出**

**100~999内的质数**

#include<stdio.h>

#include<math.h>

int main()

{

int n, i, k, count = 0;

for (n = 101; n <= 999; n += 2)

{

k = (int)sqrt(n);

for (i = 2; i <= k; i++)

{

if (n % i == 0)

{

break;

}

}

if (i > k)

{

count++;

printf("%d", n);

if (count % 8 == 0)

{

printf("\n");

}

}

}

return 0;

}

**8.** **验证歌德巴赫猜想：2000以内的正偶数（不包括2）都能够分解为两个质数之和。**

#include<stdio.h>

#include<math.h>

int judgeprime(int n);

int main()

{ int m,j,k;

int count = 1;

printf("%4d=%4d+%4d", 4, 2, 2); /\*4特殊\*/

for (m = 6; m<= 2000; m+= 2)

{

k = m / 2;/\*第一个质数应小于原偶数的一半，否则重复\*/

for (j = 3; j <= k; j += 2)

{

if (judgeprime(j) && judgeprime(m - j))/\*数学技巧，简化\*/

{

printf("%4d=%4d+%4d", m, j, m - j);

count++;

if (count % 4 == 0)/\*换行\*/

printf("\n");

break;

}

}

}

printf("\n");

return 0;

}

int judgeprime(int n)/\*返回质数\*/

{

int i, k;

if (n == 1)

return 0;

k = (int)sqrt(n);

for (i = 2; i <= k; i++)

{

if (n % i == 0)

return 0;

return n;

}

}

**9.质数判断函数**

int judgeprime(int n)

{

int i, k;

int judge = 1;

if (n == 1)

judge = 0;

k = (int)sqrt(n);

for (i = 2; i <= k; i++)

{

if (n % i == 0)

{

judge = 0;

}

}

return judge;

}

**10.利用静态局部变量求阶乘**

#include<stdio.h>

int fun(int n);

int main()

{

int i;

for (i = 1; i <= 5; i++)

{

printf("%d !=%d", i, fun(i));

}

return 0;

}

int fun(int n)

{

static int f = 1;

f = f \* n;

return f;

}

**11.** **写一个递归丽数digiSum(),输人一个非负整数，返回组成它的数字之和。例如，调digitSum(1729).则应该返回1+7+2+9.它的和是19。**

#include <stdio.h>

int digitSum(int num)

{

if (num < 10)

return num;

else

return num%10 + digitSum(num / 10);

}

int main()

{

int n, sum;

printf("Please input the number:\n");

scanf\_s("%d", &n);

sum = digitSum(n);

printf("%d\n", sum);

return 0;

}

**12.求100~999的水仙花数**

#include<stdio.h>

int narcissus(int n);

int main()

{

int i;

for (i = 100; i <= 999; i++)

{

if (narcissus(i))

{

printf("%5d", i);

}

}

printf("\n");

return 0;

}

int narcissus(int n)

{

int i, j, k;

i = n % 10;

j = n / 10 % 10;

k = n / 100;

if (i\* i\* i + j \* j \* j + k \* k \* k == n)

{

return 1;

}

else

{

return 0;

}

}

**13.统计数字出现次数**

#include<stdio.h>

int main()

{

unsigned int m, n, i;

int digit[10] = { 0 };

scanf("%u", &m); n = m;

while (m)

{

i = m % 10;

digit[i]++;

m = m / 10;

}

if (n == 0)

{

digit[0] = 1;

}

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

{

printf("%d\t", i);

}

printf("\n");

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

{

printf("%d\t", digit[i]);

}

return 0;

}

**14.求解数列中的最大数**

#include<stdio.h>

#define N 10

void printarr(int a[], int n);

int maxnum(int a[], int n);

int main()

{

int array[N], i, n;

int max, min;

do

{

printf("Please input n(1<=n<=10):\n");

scanf("%d", &n);

} while (n<1 || n>N);

printf("Please input %d elements:\n", n);

for (i = 0; i < n; i++)

{

scanf("%d", &array[i]);

}

printarr(array, n);

max = maxnum(array, n);

printf("max elements is: %d\n", max);

return 0;

}

void printarr(int a[], int n)

{

int i;

printf("The elements are :\n");

for (i = 0; i < n; i++)

printf("%5d", a[i]);

printf("\n");

}

int maxnum(int a[], int n)

{

int i, max;

max = a[0];

for(i=1;i<n;i++)

{

if (a[i] > max)

{

max = a[i];

}

}

return max;

}

**15.将3行4列矩阵转换成4行3列矩阵**

#include<stdio.h>

#include<time.h>

#include<stdlib.h>

#define ROW 3

#define COL 4

int main()

{

int a[ROW][COL];

int b[COL][ROW];

int i, j;

srand(time(NULL));

for (i = 0; i < ROW; i++)

{

for (j = 0; j < COL; j++)

{

a[i][j] = rand() % 100 + 1;

}

}

printf("Before tranpose :\n");

for (i = 0; i < ROW; i++)

{

for (j = 0; j < COL; j++)

{

printf("%4d", a[i][j]);

}

printf("\n");

}

for (i = 0; i < COL; i++)

{

for (j = 0; j < ROW; j++)

{

b[i][j] = a[j][i];

}

}

printf("After transpose:\n");

for (i = 0; i < COL; i++)

{

for (j = 0; j < ROW; j++)

{

printf("%4d", b[i][j]);

}

printf("\n");

}

return 0;

}

**16.查找元素**

#include<stdio.h>

#define SIZE 10

int find(int a[], int n, int x);

int main()

{

int a[SIZE], i = 0, n, x;

int pos;

do

{

printf("Please inpjut n(1<=n<=10):");

scanf("%d", &n);

} while (n<1 || n>SIZE);

printf("Please input %d elements:", n);

for (i = 0; i < n; i++)

{

scanf("%d", &a[i]);

}

printf("Please input x be searched:");

scanf("%d", &x);

pos = find(a, n, x);

if (pos < n)

{

printf("value=%d,index=%d", x, pos);

}

else

{

printf("Not present!\n");

}

return 0;

}

int find(int a[], int n, int x)

{

int i = 0;

while (i < n)

{

if (x == a[i])

break;

i++;

}

return i;

}

**17.插入数组元素**

#include<stdio.h>

#define SIZE 7

void print(int a[], int n);

void insert(int a[], int n, int x);

int main()

{

int x;

int a[SIZE] = { 12,13,14,15,16,17 };

print(a, SIZE - 1);

printf("Please input x be inserted");

scanf("%d", &x);

insert(a, SIZE - 1, x);

print(a, SIZE);

return 0;

}

void print(int a[], int n)

{

int i;

printf("The array is:\n");

for(i=0;i<n;i++)

{

printf("%5d", a[i]);

}

printf("\n");

}

void insert(int a[],int n,int x)

{

int i, j;

for (i = 0; i < n && a[i] < x; i++);

for (j = n - 1; j >= i; j--)

{

a[j + 1] = a[j];

}

a[i] = x;

}

**18.删除数组元素**

#include<stdio.h>

#define SIZE 5

void print(int a[], int n);

int delarray(int a[], int n, int x);

int main()

{

int x;

int a[SIZE] = { 12,13,14,15,16};

print(a, SIZE );

printf("Please input x be deleted\n");

scanf("%d", &x);

if (delarray(a, SIZE, x))

{

print(a, SIZE - 1);

}

else

{

printf("can not delete x!\n");

}

return 0;

}

void print(int a[], int n)

{

int i;

printf("The array is:\n");

for(i=0;i<n;i++)

{

printf("%5d", a[i]);

}

printf("\n");

}

int delarray(int a[], int n, int x)

{

int i, j;

int flag = 1;

for (i = 0; i < n && a[i] != x; i++);

if (i == n)

{

flag = 0;

}

else

{

for (j = i;j<n-1;j++)

{

a[j] = a[j + 1];

}

return flag;

}

**19.冒泡排序**

#include<stdio.h>

#define SIZE 10

void print(int a[], int n);

void Bubblesort(int a[], int n);

int main()

{

int a[SIZE], i, n;

do

{

printf("Please input n(1<=n<=%d", SIZE);

scanf("%d", &n);

} while (i<0 || i>SIZE);

for (i = 0; i < n; i++)

{

scanf("%d", &a[i]);

}

Bubblesort(a, n);

print(a, n);

return 0;

}

void print(int a[],int n)

{

int i;

printf("The array is:");

for (i = 0; i < n; i++)

{

printf("%5d", a[i]);

}

printf("\n");

}

void Bubblesort(int a[], int n)

{

int i, j, t;

for (i = 0; i < n - 1; i++)

{

for (j = n - 1; j > i; j--)

{

if (a[j] < a[j - 1])

{

t = a[j - 1];

a[j - 1] = a[j];

a[j] = t;

}

}

}

}

**20.选择法排序**

void sort(int a[], int n)

{

int i, k, index,t;

for (k = 0; k < n - 1; k++)

{

index = k;

for (i = 1; i < n; i++)

{

if (a[i] < index)

{

index = i;

}

}

if (index != k) **21.**

{

t = a[k];

a[k] = a[index];

a[index] = t;

}

}

}

**21.判断N阶矩阵是否对称。**

int judge(int a[][N])

{

int flag = 1,i,j;

for (i = 0; i < N; i++)

{

for (j = 0; j < i; j++)

{

if (a[i][j] != a[j][i])

{

flag = 0;

break;

}

}

}

return 0;

}

**22.用指针访问数组，求平均数**

#include<stdio.h>

int main()

{

double score[5] = { 90.5,91.5,92.5,93.5,94.5 };

double\* p = score;

int i;

double sum = 0.0;

printf("The array is:");

for (i = 0; i < 5; i++)

printf("score[%d]:\t%5.2f\t%5.2f\n", i, score[i], \*(p + i));

for (p = score; p < score + 5; p++)

{

sum += \*p;

}

return 0;

}

**23.十进制转换为二进制**

#include<stdio.h>

int main()

{

int r[16];

int\* p = r;

int m;

do

{

printf("Please input integer");

scanf("%d", &m);

} while (m < 0 || m>65535);

while (m != 0)

{

\*p = m % 2;

m / 2;

p++;

}

p--;

for (; p >= r; p--)

{

printf("%d", \*p);

}

return 0;

}

**24.统计各字符数量**

#include<stdio.h>

int main()

{

int character = 0, digit = 0, other = 0;

char t[] = "Hello!";

char\* p = t, s[20];

printf("%s\n", p);

p = s;

printf("Please input string：\n");

scanf\_s("%s", s);

while (\*p != 0)

{

if ('A' <= \*p && \*p <= 'z')

{

++character;

}

else

{

if ('0' <= \*p && \*p <= '9')

{

++digit;

}

else

{

++other;

}

}

p++;

}

printf("character:%d\n digit:%d\n other:%d\n", character, digit, other);

return 0;

}

**25.二维数组处理多个字符串**

#include<stdio.h>

int main()

{

char a[5][7];

int i;

for (i = 0; i < 5; i++)

gets\_s(a[i]);

for (i = 0; i < 5; i++)

puts(a[i]);

return 0;

}

**26.判断回文数**

#include<stdio.h>

#include<string.h>

#define MAX 80

int Palindrome(const char\* str);

int main()

{

char str[MAX], ch;

do

{

printf("Please input string:\n");

gets\_s(str);

if (Palindrome(str))

{

printf("This is a palindrom\n");

}

else

{

printf("this is not a palindrom\n");

}

printf("contine?(Yes/NO)\n");

ch = getchar();

getchar();

} while (ch != 'N' && ch != 'n');

return 0;

}

int Palindrom(const char\* str)

{

int i=0, j=strlen(str)-1;

while (i < j)

{

while (str[i] == 32)

{

i++;

}

while (str[j] == 32)

{

j--;

}

if (str[i] == str[j])

{

i++;

j--;

}

else

{

return 0;

}

}

return 1;

}

**27.统计单词个数**

int search(const char\* ps, const char\* pf)

{

int i=0, count = 0;

char dest[20];

while (\*ps)

{

dest[i++] = \*ps++;

}

dest[i] = '\n';

ps++;

if (strcmp(dest, pf) == 0)

{

count++;

}

return count;

}

**28.密码问题**

#include<stdio.h>

char passwd[] = "NJUPT";

int check(char\* ps);

int main()

{

char str[10];

int i = 0;

printf("Please input the password\n");

while ((str[i] = getchar()) != '#')

{

i++;

}

str[i] = '\n';

if (check(str))

{

printf("Pass!\n");

}

else

{

printf("Error!\n\a\a\a");

}

return 0;

}

int check(char\* ps)

{

int i = 0, flag = 1;

for (; \*ps!='\0' && flag; ps++)

{

if ('a' <= \*ps && \*ps <= 'z')

\*ps = \*ps - 32;

if (\*ps != passwd[i])

{

flag = 0;

}

else

{

i++;

}

}

return flag;

}

**29.结构体数组学生成绩排名**

#include<stdio.h>

struct student

{

int ID;

char name[20];

double score;

};

typedef struct student stu;

int input(stu s[ ]);

void sort(stu[ ], int len);

void output(const stu[], int len);

int main()

{

stu st[10];

int num;

num = input(st);

sort(st, num);

output(st, num);

return 0;

}

int input(stu s[])

{

int i, n;

do

{

printf("Enter the sum of student\n");

scanf("%d", n);

} while (n < 0 || n>10);

for (i = 0; i < n; i++)

{

printf("Enter the %d-th student", i + 1);

scanf("%d%s%lf", &s[i].ID, s[i].name, &s[i].score);

}

return n;

}

void sort(stu s[], int len)

{

int i, k, index;

stu t;

for (k = 0; k < len - 1; k++)

{

index = k;

for (i + k + 1; i < len; i++)

{

if (s[i].score > s[index].score)

index = i;

}

if (index != k)

{

t = s[index];

s[index] = s[k];

s[k] = t;

}

}

}

void output(stu s[], int len)

{

int i;

printf("学号 姓名 成绩\n");

for (i = 0; i < len; i++)

{

printf("%4d %-8s %.0f\n", s[i].ID, s[i].name, s[i].score);

}

}

**30. 输****入字符到文本文件**

#include<stdio.h>

#include<stdlib.h>

int main()

{

FILE\* fp;

char ch;

fp = fopen("D:\\cv.txt", "w");

if (fp == 0)

{

printf("file error\n");

exit(1);

}

printf("Enter a text (end with '#)");

ch = getchar();

while (ch != '#')

{

fputc(ch, fp);

ch = getchar();

}

fclose(fp);

return 0;

}

**31.读入字符到文本文件**

#include<stdio.h>

#include<stdlib.h>

int main()

{

FILE\* fp;

char ch;

fp = fopen("D:\\cv.txt", "r");

if (fp == 0)

{

printf("file error\n");

exit(1);

}

while ((ch = fgetc(fp)) != EOF)

{

putchar(ch);

}

printf("\n");

fclose(fp);

return 0;

}