

1.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

int main() {

char c;

while ((c = getchar()) != EOF) {

if (c >= 'a' && c <= 'z') {

printf("%c", c - 32);

}

else {

printf("%c", c);

}

}

system("pause");

}

2.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

#include "main.h"

#include<string.h>

typedef struct NUMBER {

char name[10];

int size;

}NUMBER;

int main() {

char c;

NUMBER num, letter, other;

strcpy(num.name, "num");

strcpy(letter.name, "alp");

strcpy(other.name, "oth");

num.size = letter.size = other.size = 0;

//统计数目

while ((c = getchar()) != EOF) {

if (c >= 'A' && c <= 'Z' || c >= 'a' && c <= 'z') {

letter.size++;

}

else if (c >= '0' && c <= '9') {

num.size++;

}

else {

other.size++;

}

}

//比大小,排柱状图

int max\_value = (num.size > letter.size ? num.size : letter.size) > other.size ? (num.size > letter.size ? num.size : letter.size) : other.size;

int min\_value = (num.size < letter.size ? num.size : letter.size) < other.size ? (num.size < letter.size ? num.size : letter.size) : other.size;

int mid\_value = 0;

//找mid\_value

if (max\_value == num.size) {

if (min\_value == letter.size) {

mid\_value = other.size;

}

else {

mid\_value = letter.size;

}

}

else if (max\_value == letter.size) {

if (min\_value == num.size) {

mid\_value = other.size;

}

else {

mid\_value = num.size;

}

}

else {//max = other

if (min\_value == num.size) {

mid\_value = letter.size;

}

else {

mid\_value = num.size;

}

}

//printf("%d\n", max\_value);

//printf("%d\n", mid\_value);

//printf("%d\n", min\_value);

//打印柱状图

for (int i = max\_value+1; i > 0; i--) {

if (i == max\_value+1) {

printf("%3d\n", max\_value);

}

else if (i < max\_value + 1 && i > mid\_value + 1) {

printf("\*\*\*\*\*\n");

}

else if (i == mid\_value + 1) {

printf("\*\*\*\*\*");

printf("\t");

printf("%3d\n", mid\_value);

}

else if (i <mid\_value + 1 && i>min\_value + 1) {

printf("\*\*\*\*\*");

printf("\t");

printf("\*\*\*\*\*");

//printf("\*\*\*\*\*");

}

else if (i == min\_value + 1) {

printf("\*\*\*\*\*");

printf("\t");

printf("\*\*\*\*\*");

printf("\t");

printf("%3d\n",min\_value);

}

else if (i < min\_value + 1 ) {

printf("\*\*\*\*\*");

printf("\t");

printf("\*\*\*\*\*");

printf("\t");

printf("\*\*\*\*\*\n");

}

//printf("\*\*\*\*\*\n");

}

//打印标识号

if (max\_value == num.size) {

if (min\_value == letter.size) {

//mid\_value = other.size;

printf(" %s \t %s \t %s\n", num.name,other.name,letter.name);

}

else {

//mid\_value = letter.size;

printf(" %s \t %s \t %s\n", num.name, letter.name,other.name);

}

}

else if (max\_value == letter.size) {

if (min\_value == num.size) {

//mid\_value = other.size;

printf(" %s \t %s \t %s\n",letter.name, other.name,num.name);

}

else {

//mid\_value = num.size;

printf(" %s \t %s \t %s\n", letter.name, num.name, other.name);

}

}

else {//max = other

if (min\_value == num.size) {

//mid\_value = letter.size;

printf(" %s \t %s \t %s\n", other.name,letter.name, num.name);

}

else {

//mid\_value = num.size;

printf(" %s \t %s \t %s\n", other.name, num.name,letter.name);

}

}

system("pause");

return 0;

}

3.1

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

#include "main.h"

#include<string.h>

typedef struct NUMBER {

char name[10];

int size;

}NUMBER;

void reverse\_print(int arr[],int length) {

//printf("%d", length);

for (int i = 0; i < length; i++) {

printf("%d", arr[length - i-1]);

}

printf("\n");

}

int main() {

int num;

int arr[35];

while (scanf("%d", &num) != EOF) {

int i = 0;

while (num) {

arr[i] = num % 2;

num = num / 2;

//printf("%d", arr[i]);

i++;

}

reverse\_print(arr,i);

}

system("pause");

return 0;

}

3.2

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

#include <math.h>

#include<string.h>

typedef struct NUMBER {

char name[10];

int size;

}NUMBER;

void reverse\_print(int arr[],int length) {

//printf("%d", length);

for (int i = 0; i < length; i++) {

printf("%d", arr[length - i-1]);

}

printf("\n");

}

int \_length(int num) {

int i = 0;

while (num) {

num = num / 10;

i++;

}

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

return i;

}

int main() {

int num;

//int dec=0;

int length = 0;

while (scanf("%d", &num) != EOF) { //1100

int dec = 0;

length = \_length(num);

for (int i = 0; i < length; i++) {

dec += (num % 10) \* (int)pow(2, i);

printf("num%10 = %d", num%10 );

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

printf("\tpow = %d\n", (int)pow(2, i));

num = num / 10;

}

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

}

system("pause");

return 0;

}

5.1

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

#include <math.h>

#include<string.h>

int main() {

int result = 0;

int arr[] = { 1,2,3,4,5,6,7,8,9,8,7,3,6,5,4,2,1 };

for (int i = 0; i < 17; i++) {

result = result ^ arr[i];

}

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

system("pause");

return 0;

}

5.2

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

#include <math.h>

#include<string.h>

int main() {

int result = 0;

int arr[] = { 1,2,3,4,5,6,7,8,9,8,7,3,6,4,2,1 }; //9 & 5

int flag = 0;

int num1, num2;

num1 = num2 = 0;

for (int i = 0; i < 16; i++) {

result = result ^ arr[i];

}

printf("result = %d\n", result);//此处为2个只出现1次的数的异或结果 12 1100

//按位的观点拆分俩数组

//

//找到异或结果数第一个不为零的位，以这个条件对数组中的元素进行分类

for (int i = 0; i < 32; i++)

{

if (((result >> i) & 1) != 1)//1100 >> 0 1 2 flag = 3

{

flag++;//用flag标记为1的位置

}

else

{

break;

}

}

for (int i = 0; i < 16; i++)

{

//二进制位第一位不为零的数

if (((arr[i] >> flag) & 1) != 1)

{

num1 ^= arr[i];

}

else

//二进制位第一位为零的数

{

num2 = num2 ^ arr[i];

}

}

printf("num1 = %d\tnum2 = %d\n", num1, num2);

system("pause");

return 0;

}