编程题参考答案（**13-15** 周）

第十三周**\_**练兵区

1. 学生成绩管理系统 **V4.0**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_LEN 10

#define STU\_NUM 30

#define COURSE\_NUM 6

int Menu(void);

/\* 字符串最大长度 \*/

/\* 最多的学生人数 \*/

/\* 最多的考试科目数 \*/

void ReadScore(long num[], char name[][MAX\_LEN], float score[][COURSE\_NUM], int n, int m);

void AverSumofEveryStudent(float score[][COURSE\_NUM], int n, int m, float sum[STU\_NUM],

float aver[STU\_NUM]);

void AverSumofEveryCourse(float score[][COURSE\_NUM], int n, int m);

void SortbyScore(long num[], char name[][MAX\_LEN], float score[][COURSE\_NUM], float sum[],

float aver[], int n, int m, int (\*compare)(float a, float b));

int

int

Ascending(float a, float b);

Descending(float a, float b);

void SwapFloat(float \*x, float \*y);

void SwapLong(long \*x, long \*y);

void SwapChar(char x[], char y[]);

void AsSortbyNum(long num[], char name[][MAX\_LEN], float score[][COURSE\_NUM], float

sum[], float aver[], int n, int m);

void SortbyName(long num[], char name[][MAX\_LEN], float score[][COURSE\_NUM], float

sum[], float aver[], int n, int m);

void SearchbyNum(long num[], char name[][MAX\_LEN], float score[][COURSE\_NUM], float

sum[], float aver[], int n, int m);

void SearchbyName(long num[], char name[][MAX\_LEN], float score[][COURSE\_NUM], float

sum[], float aver[], int n, int m);

void StatisticAnalysis(float score[][COURSE\_NUM], int n, int m);

void PrintScore(long num[], char name[][MAX\_LEN], float score[][COURSE\_NUM], float sum[],

float aver[], int n, int m) ;

int main()

{

char ch;

int

n = 0, m = 0;

float score[STU\_NUM][COURSE\_NUM], sum[STU\_NUM], aver[STU\_NUM];

long num[STU\_NUM];

char name[STU\_NUM][MAX\_LEN];

printf("Input student number(n<30):\n", STU\_NUM);

scanf("%d", &n);

while (1)

{

ch = Menu();

switch (ch)

{

/\* 显示菜单，并读取用户输入 \*/

case 1: printf("Input course number(m<=%d):\n",COURSE\_NUM);

scanf("%d", &m);

ReadScore(num, name, score, n, m);

break;

case 2: AverSumofEveryCourse(score, n, m);

break;

case 3: AverSumofEveryStudent(score, n, m, sum, aver);

break;

case 4: SortbyScore(num,name,score,sum,aver,n,m,Descending);

printf("Sort in descending order by score:\n");

PrintScore(num, name, score, sum, aver, n, m);

break;

case 5: SortbyScore(num,name,score,sum,aver,n,m,Ascending);

printf("Sort in ascending order by score:\n");

PrintScore(num, name, score, sum, aver, n, m);

break;

case 6: AsSortbyNum(num, name, score, sum, aver, n, m);

printf("Sort in ascending order by number:\n");

PrintScore(num, name, score, sum, aver, n, m);

break;

case 7: SortbyName(num, name, score, sum, aver, n, m);

printf("Sort in dictionary order by name:\n");

PrintScore(num, name, score, sum, aver, n, m);

break;

case 8: SearchbyNum(num, name, score, sum, aver, n, m);

break;

case 9: SearchbyName(num, name, score, sum, aver, n, m);

break;

case 10:StatisticAnalysis(score, n, m);

break;

case 11:PrintScore(num, name, score, sum, aver, n, m);

break;

case 0: printf("End of program!");

exit(0);

default:printf("Input error!\n");

}

}

return 0;

}

/\* 函数功能：显示菜单并获得用户键盘输入的选项 \*/

int Menu(void)

{

int itemSelected;

printf("Management for Students' scores\n");

printf("1.Input record\n");

printf("2.Caculate total and average score of every course\n");

printf("3.Caculate total and average score of every student\n");

printf("4.Sort in descending order by score\n");

printf("5.Sort in ascending order by score\n");

printf("6.Sort in ascending order by number\n");

printf("7.Sort in dictionary order by name\n");

printf("8.Search by number\n");

printf("9.Search by name\n");

printf("10.Statistic analysis\n");

printf("11.List record\n");

printf("0.Exit\n");

printf("Please Input your choice:\n");

scanf("%d", &itemSelected); /\* 读入用户输入 \*/

return itemSelected;

}

/\* 函数功能：输入 n 个学生的 m 门课成绩 \*/

void ReadScore(long num[], char name[][MAX\_LEN],

float score[][COURSE\_NUM], int n, int m)

{

int i, j;

printf("Input student's ID, name and score:\n");

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

{

scanf("%ld%s", &num[i], name[i]);

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

{

scanf("%f", &score[i][j]);

}

}

}

/\* 函数功能：计算每个学生各门课程的总分和平均分 \*/

void AverSumofEveryStudent(float score[][COURSE\_NUM], int n, int m, float sum[STU\_NUM],

float aver[STU\_NUM])

{

int i, j;

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

{

sum[i] = 0;

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

{

}

sum[i] = sum[i] + score[i][j];

aver[i] = m>0 ? sum[i] / m : -1;

printf("student %d:sum=%.0f,aver=%.0f\n",i+1,sum[i],aver[i]);

}

}

/\* 函数功能：计算每门课程的总分和平均分 \*/

void AverSumofEveryCourse(float score[][COURSE\_NUM], int n, int m)

{

int i, j;

float sum[COURSE\_NUM], aver[COURSE\_NUM];

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

{

sum[j] = 0;

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

{

sum[j] = sum[j] + score[i][j];

}

aver[j] = n>0 ? sum[j] / n : -1;

printf("course %d:sum=%.0f,aver=%.0f\n",j+1,sum[j],aver[j]);

}

}

/\* 函数功能：按选择法将数组 sum 的元素值排序 \*/

void SortbyScore(long num[], char name[][MAX\_LEN], float score[][COURSE\_NUM], float sum[],

float aver[], int n, int m, int (\*compare)(float a, float b))

{

int i, j, k, t;

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

{

k = i;

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

{

if ((\*compare)(sum[j], sum[k]))

k = j;

}

if (k != i)

{

for (t=0; t<m; t++)

/\* 交换 m 门课程的成绩 \*/

{

SwapFloat(&score[k][t], &score[i][t]);

}

SwapFloat(&sum[k], &sum[i]);

/\* 交换总分 \*/

SwapFloat(&aver[k], &aver[i]); /\* 交换平均分 \*/

SwapLong(&num[k], &num[i]); /\* 交换学号 \*/

SwapChar(name[k], name[i]);

/\* 交换姓名 \*/

}

}

}

/\* 使数据按升序排序 \*/

int Ascending(float a, float b)

{

return a < b;

/\* 这样比较决定了按升序排序，如果 a<b，则交换 \*/

}

/\* 使数据按降序排序 \*/

int Descending(float a, float b)

{

return a > b;

/\* 这样比较决定了按降序排序，如果 a>b，则交换 \*/

}

/\* 交换两个单精度浮点型数据 \*/

void SwapFloat(float \*x, float \*y)

{

float temp;

temp = \*x;

\*x = \*y;

\*y = temp;

}

/\* 交换两个长整型数据 \*/

void SwapLong(long \*x, long \*y)

{

long temp;

temp = \*x;

\*x = \*y;

\*y = temp;

}

/\* 交换两个字符串 \*/

void SwapChar(char x[], char y[])

{

char temp[MAX\_LEN];

strcpy(temp, x);

strcpy(x, y);

strcpy(y, temp);

}

/\* 函数功能：按选择法将数组 num 的元素值按从低到高排序 \*/

void AsSortbyNum(long num[], char name[][MAX\_LEN], float score[][COURSE\_NUM], float

sum[], float aver[], int n, int m)

{

int i, j, k, t;

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

{

k = i;

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

{

if (num[j] < num[k])k = j;

}

if (k != i)

{

for (t=0; t<m; t++)

{

/\* 交换 m 门课程的成绩 \*/

/\* 交换总分 \*/

SwapFloat(&score[k][t], &score[i][t]);

}

SwapFloat(&sum[k], &sum[i]);

SwapFloat(&aver[k], &aver[i]); /\* 交换平均分 \*/

SwapLong(&num[k], &num[i]);

/\* 交换学号 \*/

/\* 交换姓名 \*/

SwapChar(name[k], name[i]);

}

}

}

/\* 函数功能：交换法实现字符串按字典顺序排序 \*/

void SortbyName(long num[], char name[][MAX\_LEN], float score[][COURSE\_NUM],

float sum[], float aver[], int n, int m)

{

int i, j, t;

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

{

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

{

if (strcmp(name[j], name[i]) < 0)

{

for (t=0; t<m; t++)

{

/\* 交换 m 门课程的成绩 \*/

SwapFloat(&score[i][t], &score[j][t]);

}

SwapFloat(&sum[i], &sum[j]);

/\* 交换总分 \*/

SwapFloat(&aver[i], &aver[j]); /\* 交换平均分 \*/

SwapLong(&num[i], &num[j]);

SwapChar(name[i], name[j]);

/\* 交换学号 \*/

/\* 交换姓名 \*/

}

}

}

}

/\* 函数功能：按学号查找学生成绩并显示查找结果 \*/

void SearchbyNum(long num[], char name[][MAX\_LEN], float score[][COURSE\_NUM], float

sum[], float aver[], int n, int m)

{

long number;

int

i, j;

printf("Input the number you want to search:\n");

scanf("%ld", &number);

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

{

if (num[i] == number)

{

printf("%ld\t%s\t", num[i], name[i]);

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

{

printf("%.0f\t", score[i][j]);

}

printf("%.0f\t%.0f\n", sum[i], aver[i]);

return;

}

}

printf("Not found!\n");

}

/\* 函数功能：按姓名的字典顺序排出成绩表 \*/

void SearchbyName(long num[], char name[][MAX\_LEN], float score[][COURSE\_NUM], float

sum[], float aver[], int n, int m)

{

char x[MAX\_LEN];

int i, j;

printf("Input the name you want to search:\n");

scanf("%s", x);

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

{

if (strcmp(name[i], x) == 0)

{

printf("%ld\t%s\t", num[i], name[i]);

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

{

printf("%.0f\t", score[i][j]);

}

printf("%.0f\t%.0f\n", sum[i], aver[i]);

return;

}

}

printf("Not found!\n");

}

/\* 函数功能：统计各分数段的学生人数及所占的百分比 \*/

void StatisticAnalysis(float score[][COURSE\_NUM], int n, int m)

{

int i, j, total, t[6];

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

{

printf("For course %d:\n", j+1);

memset(t, 0, sizeof(t));

/\* 将数组 t 的全部元素初始化为 0 \*/

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

{

if (score[i][j]>=0 && score[i][j]<60) t[0]++;

else if (score[i][j]<70)

else if (score[i][j]<80)

else if (score[i][j]<90)

else if (score[i][j]<100)

else if (score[i][j] == 100)

t[1]++;

t[2]++;

t[3]++;

t[4]++;

t[5]++;

}

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

{

total = total + t[i];

}

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

{

if (i == 0) printf("<60\t%d\t%.2f%%\n",t[i],(float)t[i]/n\*100);

else if (i == 5) printf("%d\t%d\t%.2f%%\n", (i+5)\*10,t[i],(float)t[i]/n\*100);

else

printf("%d-%d\t%d\t%.2f%%\n", (i+5)\*10, (i+5)\*10+9, t[i], (float)t[i]/n\*100);

}

}

}

/\* 函数功能： 打印学生成绩 \*/

void PrintScore(long num[], char name[][MAX\_LEN], float score[][COURSE\_NUM], float sum[],

float aver[], int n, int m)

{

int i, j;

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

{

printf("%ld\t%s\t", num[i], name[i]);

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

{

printf("%.0f\t", score[i][j]);

}

printf("%.0f\t%.0f\n", sum[i], aver[i]);

}

}

**2.** 寻找最高分成绩的学生

#include <stdio.h>

#include <stdlib.h>

void InputScore(int \*p, int m, int n);

int FindMax(int \*p, int m, int n, int \*pRow, int \*pCol);

int main()

{

int \*pScore, m, n, maxScore, row, col;

printf("Input array size m,n:\n");

scanf("%d,%d", &m, &n);

pScore = (int \*)calloc(m\*n, sizeof (int)); /\* 申请动态内存 \*/

if (pScore == NULL)

{

printf("No enough memory!\n");

exit(0);

}

InputScore(pScore, m, n);

maxScore = FindMax(pScore, m, n, &row, &col);

printf("maxScore = %d, class = %d, number = %d\n", maxScore, row+1, col+1);

free(pScore);

return 0;

/\* 释放动态内存 \*/

}

/\* 函数功能：输入 m 行 n 列二维数组的值 \*/

void InputScore(int \*p, int m, int n)

{

int i, j;

printf("Input %d\*%d array:\n", m, n);

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

{

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

{

scanf("%d", &p[i\*n+j]);

}

}

}

/\* 函数功能：计算任意 m 行 n 列二维数组中元素的最大值，并指出其所在行列下标值 \*/

int FindMax(int \*p, int m, int n, int \*pRow, int \*pCol)

{

int i, j, max = p[0];

\*pRow = 0;

\*pCol = 0;

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

{

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

{

if (p[i\*n+j] > max)

{

max = p[i\*n+j];

\*pRow = i;

/\*记录行下标\*/

/\*记录列下标\*/

\*pCol = j;

}

}

}

return max;

}

**3.** 程序改错

#include <stdio.h>

#define STUD 30

#define COURSE 5

/\* 最多可能的学生人数 \*/

/\* 最多可能的考试科目数 \*/

void Total(int \*pScore, int sum[], float aver[], int m, int n);

void Print(int \*pScore, int sum[], float aver[], int m, int n);

int main()

{

int

i, j, m, n, score[STUD][COURSE], sum[STUD];

float aver[STUD];

printf("How many students?\n");

scanf("%d", &m);

printf("How many courses?\n");

scanf("%d", &n);

printf("Input scores:\n");

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

{

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

{

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

}

}

Total(\*score, sum, aver, m, n);

Print(\*score, sum, aver, m, n);

return 0;

}

void Total(int \*pScore, int sum[], float aver[], int m, int n)

{

int i, j;

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

{

sum[i] = 0;

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

{

sum[i] = sum[i] + pScore[i\* COURSE + j];

}

aver[i] = (float) sum[i] / n;

}

}

void Print(int \*pScore, int sum[], float aver[], int m, int n)

{

int i, j;

printf("Result:\n");

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

{

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

{

printf("%4d", pScore[i\* COURSE + j]);

}

printf("%5d%6.1f\n", sum[i], aver[i]);

}

}

**4.** 矩阵转置

#include <stdio.h>

#define M 10

#define N 10

void Transpose(int \*a, int \*at, int m, int n);

void InputMatrix(int \*a, int m, int n);

void PrintMatrix(int \*at, int n, int m);

int main()

{

int s[M][N], st[N][M], m, n;

printf("Input m, n:\n");

scanf("%d,%d", &m, &n);

InputMatrix(\*s, m, n);

Transpose(\*s, \*st, m, n);

printf("The transposed matrix is:\n");

PrintMatrix(\*st, n, m);

return 0;

}

/\* 函数功能：计算 m\*n 矩阵 a 的转置矩阵 at \*/

void Transpose(int \*a, int \*at, int m, int n)

{

int i, j;

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

{

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

{

at[j\*m+i] = a[i\*n+j];

}

}

}

/\* 函数功能：输入 m\*n 矩阵 a 的值 \*/

void InputMatrix(int \*a, int m, int n)

{

int i, j;

printf("Input %d\*%d matrix:\n", m, n);

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

{

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

{

scanf("%d", &a[i\*n+j]);

}

}

}

/\* 函数功能：输出 n\*m 矩阵 at 的值 \*/

void PrintMatrix(int \*at, int n, int m)

{

int i, j;

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

{

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

{

printf("%d\t", at[i\*m+j]);

}

printf("\n");

}

}

**5.** 在升序排序的数组中插入一个元素

#include<stdio.h>

#define N 20

/\* 插入前数组最大元素个数 \*/

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

int main()

{

int a[N+1];

int x, i, n;

/\* 定义数组长度为插入前的数组元素个数加 1 \*/

/\* 输入插入前数组元素个数 \*/

printf("Input array size:\n");

scanf("%d", &n);

printf("Input array:\n");

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

{

scanf("%d", &a[i]); /\* 输入插入前已按升序排序的数组元素 \*/

}

printf("Input x:\n");

scanf("%d", &x);

Insert(a, n, x);

/\* 输入待插入的元素 x \*/

/\* 插入元素 x 到已排序数组中 \*/

printf("After insert %d:\n", x);

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

{

printf("%4d", a[i]); /\* 输出插入 x 后的数组元素 \*/

}

return 0;

}

/\* 函数功能：将 x 插入到一个已按升序排序的数组中 \*/

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

{

int i = 0, pos;

while (i < n && x > a[i]) /\* 查找待插入位置 \*/

{

i++;

}

pos = i;

/\* 记录元素 x 应插入的数组下标位置 pos \*/

for (i = n-1; i>= pos; i--)/\* 从尾部开始移动 pos 及其后所有的元素 \*/

{

a[i+1] = a[i];

/\* 向后复制数组元素 \*/

}

a[pos] = x;

/\* 插入元素 x 到位置 pos \*/

}

**6.** 计算平均数、中位数和众数

#include <stdio.h>

#define

#define

M

N

40

11

int Mean(int answer[], int n);

int Median(int answer[], int n);

int Mode(int answer[], int n);

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

int main()

{

int i, feedback[M];

printf("Input the feedbacks of 40 students:\n");

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

{

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

}

printf("Mean value=%d\n", Mean(feedback, M));

printf("Median value=%d\n", Median(feedback, M));

printf("Mode value=%d\n", Mode(feedback, M));

return 0;

}

/\* 函数功能：若 n>0 则计算并返回 n 个数的平均数，否则返回-1 \*/

int Mean(int answer[], int n)

{

int i, sum = 0;

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

{

sum += answer[i];

}

return n>0 ? sum/n : -1;

}

/\* 函数功能：计算 n 个数的中位数 \*/

int Median(int answer[], int n)

{

DataSort(answer, n);

if (n%2 == 0)

return (answer[n/2] + answer[n/2-1]) / 2;

else

return answer[n/2];

}

/\* 函数功能：计算 n 个数的众数 \*/

int Mode(int answer[], int n)

{

int i, grade, max = 0, modeValue = 0, count[N] = {0};

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

{

count[answer[i]]++;

}

for (grade=1; grade<=N-1; grade++)

{

if (count[grade] > max)

{

max = count[grade];

modeValue = grade;

}

}

return modeValue;

}

/\* 函数功能：按选择法对数组 a 中的 n 个元素进行排序 \*/

void DataSort(int a[], int n)

{

int i, j, k, temp;

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

{

k = i;

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

{

if (a[j] > a[k]) k = j;

}

if (k != i)

{

temp = a[k];

a[k] = a[i];

a[i] = temp;

}

}

}

第十四周**\_**练兵区

**1.** 计算零件数

#include <stdio.h>

int main()

{

int i;

for (i = 100; i <= 200; i++)

{

if ((i - 2) % 4 == 0)

{

if ((i - 3) % 7 == 0)

{

if ((i - 5) % 9 == 0)

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

}

}

}

return 0;

}

2. 走台阶

#include<stdio.h>

int main()

{

int i = 0, a[10];

a[0] = 1;

a[1] = 2;

for (i = 2; i < 10; ++i)

{

a[i] = a[i - 1] + a[i - 2];

}

printf("Result=%d", a[9]);

return 0;

}

**3.** 将数据按照奇偶排序

#include "stdio.h"

int main()

{

int arr[10], brr[10];

int i, j, tmp;

printf("Input 10 numbers:\n");

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

{

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

}

for (i = 0; i < 9; i++)

{

for (j = i + 1; j < 10; j++)

{

if (arr[j] < arr[i])

{

tmp = arr[i];

arr[i] = arr[j];

arr[j] = tmp;

}

}

}

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

{

if (arr[i] % 2 == 1)

{

brr[j++] = arr[i];

}

}

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

{

if (arr[i] % 2 == 0)

{

brr[j++] = arr[i];

}

}

printf("Output: ");

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

{

if (i < 9)

{

printf("%d,", brr[i]);

}

else

{

printf("%d\n", brr[i]);

}

}

}

**4.** 三色球分组

#include <stdio.h>

int Fun(void);

int main()

{

int sum;

sum = Fun();

printf("sum=%4d\n", sum);

return 0;

}

int Fun(void)

{

int i, j, k, sum = 0;

printf("The result:\n");

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

{

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

{

for (k = 0; k <= 6; k++)

{

if (i + j + k == 8)

{

printf("red:%4d white:%4d black:%4d\n", i, j, k);

sum = sum + 1;

}

}

}

}

return sum;

}

**5.** 同构数

#include <stdio.h>

int main()

{

int m;

for (m = 1; m <= 99; m++)

{

if (m\*m % 10 == m || m\*m % 100 == m)

{

printf("m=%3d\t\tm\*m=%6d\n", m, m\*m);

}

}

return 0;

}

第十五周**\_**练兵区

**1.** 学生成绩管理系统 **V5.0**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_LEN 10

#define STU\_NUM 30

#define COURSE\_NUM 6

typedef struct student

{

/\* 字符串最大长度 \*/

/\* 最多的学生人数 \*/

/\* 最多的考试科目数 \*/

long num;

/\* 每个学生的学号 \*/

/\* 每个学生的姓名 \*/

char name[MAX\_LEN];

float score[COURSE\_NUM];

float sum;

/\* 每个学生 COURSE\_NUM 门功课的成绩 \*/

/\* 每个学生的总成绩 \*/

float aver;

/\* 每个学生的平均成绩 \*/

}STU;

int

Menu(void);

void ReadScore(STU stu[], int n, int m);

void AverSumofEveryStudent(STU stu[], int n, int m);

void AverSumofEveryCourse(STU stu[], int n, int m);

void SortbyScore(STU stu[],int n,int m,int (\*compare)(float a,float b));

int

int

Ascending(float a, float b);

Descending(float a, float b);

void SwapFloat(float \*x, float \*y);

void SwapLong(long \*x, long \*y);

void SwapChar(char x[], char y[]);

void AsSortbyNum(STU stu[], int n, int m);

void SortbyName(STU stu[], int n, int m);

void SearchbyNum(STU stu[], int n, int m);

void SearchbyName(STU stu[], int n, int m);

void StatisticAnalysis(STU stu[], int n, int m);

void PrintScore(STU stu[], int n, int m);

int main()

{

char ch;

int

n = 0, m = 0; /\* 学生人数为 n，课程门数为 m \*/

STU

stu[STU\_NUM];

printf("Input student number(n<=30):\n", STU\_NUM);

scanf("%d", &n);

while (1)

{

ch = Menu();

switch (ch)

{

/\* 显示菜单，并读取用户输入 \*/

case 1:

printf("Input course number(m<=%d):\n",COURSE\_NUM);

scanf("%d", &m);

ReadScore(stu, n, m);

break;

case 2: AverSumofEveryCourse(stu, n, m);

break;

case 3: AverSumofEveryStudent(stu, n, m);

break;

case 4: SortbyScore(stu, n, m, Descending);

printf("Sort in descending order by score:\n");

PrintScore(stu, n, m);

break;

case 5: SortbyScore(stu, n, m, Ascending);

printf("Sort in ascending order by score:\n");

PrintScore(stu, n, m);

break;

case 6: AsSortbyNum(stu, n, m);

printf("Sort in ascending order by number:\n");

PrintScore(stu, n, m);

break;

case 7: SortbyName(stu, n, m);

printf("Sort in dictionary order by name:\n");

PrintScore(stu, n, m);

break;

case 8: SearchbyNum(stu, n, m);

break;

case 9: SearchbyName(stu, n, m);

break;

case 10: StatisticAnalysis(stu, n, m);

break;

case 11:PrintScore(stu, n, m);

break;

case 0: printf("End of program!");

exit(0);

default:printf("Input error!\n");

}

}

return 0;

}

/\* 函数功能：显示菜单并获得用户键盘输入的选项 \*/

int Menu(void)

{

int itemSelected;

printf("Management for Students' scores\n");

printf("1.Input record\n");

printf("2.Caculate total and average score of every course\n");

printf("3.Caculate total and average score of every student\n");

printf("4.Sort in descending order by score\n");

printf("5.Sort in ascending order by score\n");

printf("6.Sort in ascending order by number\n");

printf("7.Sort in dictionary order by name\n");

printf("8.Search by number\n");

printf("9.Search by name\n");

printf("10.Statistic analysis\n");

printf("11.List record\n");

printf("0.Exit\n");

printf("Please Input your choice:\n");

scanf("%d", &itemSelected); /\* 读入用户输入 \*/

return itemSelected;

}

/\* 函数功能：输入 n 个学生的 m 门课成绩 \*/

void ReadScore(STU stu[], int n, int m)

{

int i, j;

printf("Input student's ID, name and score:\n");

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

{

scanf("%ld%s", &stu[i].num, stu[i].name);

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

{

scanf("%f", &stu[i].score[j]);

}

}

}

/\* 函数功能：计算每个学生各门课程的总分和平均分 \*/

void AverSumofEveryStudent(STU stu[], int n, int m)

{

int i, j;

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

{

stu[i].sum = 0;

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

{

stu[i].sum = stu[i].sum + stu[i].score[j];

}

stu[i].aver = m>0 ? stu[i].sum / m : -1;

printf("student %d: sum=%.0f,aver=%.0f\n", i+1, stu[i].sum, stu[i].aver);

}

}

/\* 函数功能：计算每门课程的总分和平均分 \*/

void AverSumofEveryCourse(STU stu[], int n, int m)

{

int

i, j;

float sum[COURSE\_NUM], aver[COURSE\_NUM];

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

{

sum[j] = 0;

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

{

sum[j] = sum[j] + stu[i].score[j];

}

aver[j] = n>0 ? sum[j] / n : -1;

printf("course %d:sum=%.0f,aver=%.0f\n", j+1,sum[j],aver[j]);

}

}

/\* 函数功能：按选择法将数组 sum 的元素值排序 \*/

void SortbyScore(STU stu[], int n, int m, int (\*compare)(float a, float b))

{

int i, j, k, t;

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

{

k = i;

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

{

if ((\*compare)(stu[j].sum, stu[k].sum)) k = j;

}

if (k != i)

{

for (t=0; t<m; t++)

{

/\* 交换 m 门课程的成绩 \*/

SwapFloat(&stu[k].score[t], &stu[i].score[t]);

}

SwapFloat(&stu[k].sum, &stu[i].sum);

/\* 交换总分 \*/

SwapFloat(&stu[k].aver, &stu[i].aver); /\* 交换平均分 \*/

SwapLong(&stu[k].num, &stu[i].num);

SwapChar(stu[k].name, stu[i].name);

/\* 交换学号 \*/

/\* 交换姓名 \*/

}

}

}

/\* 使数据按升序排序 \*/

int Ascending(float a, float b)

{

return a < b;

/\* 这样比较决定了按升序排序，如果 a<b，则交换 \*/

}

/\* 使数据按降序排序 \*/

int Descending(float a, float b)

{

}

return a > b;

/\* 这样比较决定了按降序排序，如果 a>b，则交换 \*/

/\* 交换两个单精度浮点型数据 \*/

void SwapFloat(float \*x, float \*y)

{

float temp;

temp = \*x;

\*x = \*y;

\*y = temp;

}

/\* 交换两个长整型数据 \*/

void SwapLong(long \*x, long \*y)

{

long temp;

temp = \*x;

\*x = \*y;

\*y = temp;

}

/\* 交换两个字符串 \*/

void SwapChar(char x[], char y[])

{

char temp[MAX\_LEN];

strcpy(temp, x);

strcpy(x, y);

strcpy(y, temp);

}

/\* 函数功能：按选择法将数组 num 的元素值按从低到高排序 \*/

void AsSortbyNum(STU stu[], int n, int m)

{

int i, j, k, t;

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

{

k = i;

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

{

if (stu[j].num < stu[k].num) k = j;

}

if (k != i)

{

for (t=0; t<m; t++)

{

/\* 交换 m 门课程的成绩 \*/

SwapFloat(&stu[k].score[t], &stu[i].score[t]);

}

SwapFloat(&stu[k].sum, &stu[i].sum);

/\* 交换总分 \*/

SwapFloat(&stu[k].aver, &stu[i].aver); /\* 交换平均分 \*/

SwapLong(&stu[k].num, &stu[i].num);

/\* 交换学号 \*/

/\* 交换姓名 \*/

SwapChar(stu[k].name, stu[i].name);

}

}

}

/\* 函数功能：交换法实现字符串按字典顺序排序 \*/

void SortbyName(STU stu[], int n, int m)

{

int i, j, t;

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

{

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

{

if (strcmp(stu[j].name, stu[i].name) < 0)

{

for (t=0; t<m; t++) /\* 交换 m 门课程的成绩 \*/

{

SwapFloat(&stu[i].score[t], &stu[j].score[t]);

}

SwapFloat(&stu[i].sum, &stu[j].sum);

/\* 交换总分 \*/

SwapFloat(&stu[i].aver, &stu[j].aver); /\* 交换平均分 \*/

SwapLong(&stu[i].num, &stu[j].num);

SwapChar(stu[i].name, stu[j].name);

/\* 交换学号 \*/

/\* 交换姓名 \*/

}

}

}

}

/\* 函数功能：按学号查找学生成绩并显示查找结果 \*/

void SearchbyNum(STU stu[], int n, int m)

{

long number;

int

i, j;

printf("Input the number you want to search:\n");

scanf("%ld", &number);

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

{

if (stu[i].num == number)

{

printf("%ld\t%s\t", stu[i].num, stu[i].name);

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

{

printf("%.0f\t", stu[i].score[j]);

}

printf("%.0f\t%.0f\n", stu[i].sum, stu[i].aver);

return;

}

}

printf("Not found!\n");

}

/\* 函数功能：按姓名的字典顺序排出成绩表 \*/

void SearchbyName(STU stu[], int n, int m)

{

char x[MAX\_LEN];

int i, j;

printf("Input the name you want to search:\n");

scanf("%s", x);

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

{

if (strcmp(stu[i].name, x) == 0)

{

printf("%ld\t%s\t", stu[i].num, stu[i].name);

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

{

printf("%.0f\t", stu[i].score[j]);

}

printf("%.0f\t%.0f\n", stu[i].sum, stu[i].aver);

return;

}

}

printf("Not found!\n");

}

/\* 函数功能：统计各分数段的学生人数及所占的百分比 \*/

void StatisticAnalysis(STU stu[], int n, int m)

{

int i, j, total, t[6];

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

{

printf("For course %d:\n", j+1);

memset(t, 0, sizeof(t));

/\* 将数组 t 的全部元素初始化为 0 \*/

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

{

if (stu[i].score[j]>=0 && stu[i].score[j]<60) t[0]++;

else if (stu[i].score[j]<70)

else if (stu[i].score[j]<80)

else if (stu[i].score[j]<90)

t[1]++;

t[2]++;

t[3]++;

else if (stu[i].score[j]<100) t[4]++;

else if (stu[i].score[j] == 100)

t[5]++;

}

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

{

total = total + t[i];

}

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

{

if (i==0) printf("<60\t%d\t%.2f%%\n",t[i],(float)t[i]/n\*100);

else if (i==5) printf("%d\t%d\t%.2f%%\n", (i+5)\*10,t[i],(float)t[i]/n\*100);

else printf("%d-%d\t%d\t%.2f%%\n", (i+5)\*10, (i+5)\*10+9, t[i], (float)t[i]/n\*100);

}

}

}

/\* 函数功能： 打印学生成绩 \*/

void PrintScore(STU stu[], int n, int m)

{

int i, j;

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

{

printf("%ld\t%s\t", stu[i].num, stu[i].name);

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

{

printf("%.0f\t", stu[i].score[j]);

}

printf("%.0f\t%.0f\n", stu[i].sum, stu[i].aver);

}

}

**2.** 字符串中的字符排序

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define N 20

void mySort(char str[], int n)

{

int i, j, tmp;

if (str == NULL || n <= 0)

return;

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

{

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

{

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

{

tmp = str[i];

str[i] = str[j];

str[j] = tmp;

}

}

}

}

int main()

{

char str[N];

int n;

printf("Input a string:\n");

gets(str);

n = strlen(str);

mySort(str, n);

printf("%s", str);

return 0;

}

**3.** 纯数字字符串检验

#include<stdio.h>

int IsAllDigit(char p[]);

int main()

{

char a[80];

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

gets(a);

if (IsAllDigit(a))

printf("The string is digit string.");

else

printf("The string is not digit string.");

}

int IsAllDigit(char p[])

{

int flag = 1, i;

for (i = 0; p[i] != '\0'; i++)

{

if (p[i] < '0' || p[i] > '9')

{

flag = 0;

break;

}

}

return flag;

}

**4.** 孪生素数

#include<stdio.h>

#include<math.h>

int IsPrime(int x);

int main()

{

long c, d, i, f = 0;

int n = 0;

printf("please input c,d(c>2):\n");

scanf("%ld,%ld", &c, &d);

if (c%2 == 0)

{

c++;

}

for (i=c; i<=d; i+=2)

{

if (IsPrime(i))

{

if (i-f == 2)

{

printf("(%ld,%ld)\n", f, i);

n++;

}

f = i;

}

}

printf("total=%d\n", n);

return 0;

}

// 函数功能：判断 x 是否是素数，若函数返回 0，则表示不是素数，若返回 1，则代表是

int IsPrime(int x)

{

int i, flag = 1;

int squareRoot = (int)sqrt(x);

if (x <= 1) flag = 0;

// 负数、0 和 1 都不是素数

for (i=2; i<=squareRoot && flag; i++)

{

if (x%i == 0) flag = 0; // 若能被整除，则不是素数

}

}

return flag;

**5.** 求解不等式

#include <stdio.h>

#include <math.h>

int main()

{

int i, m;

double s, n;

printf("Input n:\n");

scanf("%lf", &n);

for (m = 1; m <= 10000; m++)

{

s = 0;

for (i = m; i <= 2 \* m; i++)

{

s = s + sqrt(i);

}

if (s > n)

{

break;

}

}

printf("Result:m>=%d\n", m);

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

return 0;

}

**6.** 梅森尼数

#include<stdio.h>

#include<math.h>

int IsPrime(double x);

int main()

{

double t, m;

int

count = 0, i, n;

printf("Input n:\n");

scanf("%d", &n);

t = 2;

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

{

t = t \* 2;

m

= t - 1;

if (IsPrime(m))

{

count++;

printf("2^%d-1=%.0lf\n", i, m);

}

}

printf("count=%d\n", count);

return 0;

}

// 函数功能：判断 x 是否是素数，若函数返回 0，则表示不是素数，若返回 1，则代表是素

数

int IsPrime(double x)

{

int i, flag = 1;

int squareRoot = (int)sqrt(x);

if (x <= 1) flag = 0;

// 负数、0 和 1 都不是素数

for (i=2; i<=squareRoot && flag; i++)

{

if (x/i == (int)(x/i)) flag = 0; // 若能被整除，则不是素数

}

return flag;

}

**7.** 回文素数

#include<stdio.h>

#include<math.h>

int IsPrime(int x);

int main()

{

int i, j, k, t, m, n, count = 0;

printf("Input n:\n");

scanf("%d", &n);

for (m=10; m<n; m++)

{

// 从 10 开始试到 n-1

i = m / 100;

// 分离出百位数字

// 分离出十位数字

j = (m - i \* 100) / 10;

k = m % 10;

// 分离出个位数字

if (m < 100)

// 若为两位数

//右读结果

// 若为三位数

t = k \* 100 + j \* 10 + i; //右读结果

{

t = k \* 10 + j ;

}

else

{

}

if (m==t && IsPrime(m))

{

printf("%4d", m);

count++;

}

}

printf("\ncount=%d\n", count);

return 0;

}

// 函数功能：判断 x 是否是素数，若函数返回 0，则表示不是素数，若返回 1，则代表是素

数

int IsPrime(int x)

{

int i, flag = 1;

int squareRoot = (int)sqrt(x);

if (x <= 1) flag = 0;

// 负数、0 和 1 都不是素数

for (i=2; i<=squareRoot && flag; i++)

{

if (x%i == 0) flag = 0; // 若能被整除，则不是素数

}

return flag;

}

**8.** 完全数

#include <stdio.h>

#include <math.h>

int IsPerfect(int x);

void OutputFactor(int m);

int main()

{

int m;

printf("Input m:\n");

scanf("%d", &m);

if (IsPerfect(m)) //若 m 是完全数

{

printf("Yes!\n");

OutputFactor(m);

printf("No!\n");

}

else

{

//若 m 不是完全数

}

return 0;

}

// 函数功能：判断完全数，若函数返回 0，则代表不是完全数，若返回 1，则代表是完全数

int IsPerfect(int x)

{

int i;

int sum = 0; //x 为 1 时，sum=0，函数将返回 0，表示 1 没有真因子，不是完全数

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

{

if (x%i == 0)

{

sum = sum + i;

}

}

return sum==x ? 1 : 0;

}

// 函数功能：输出 x 的所有包括 1 在内的因子

void OutputFactor(int m)

{

int i, isFirstFactor = 1;

for (i=1; i<fabs(m); i++)//输出包括 1 在内的因子，所以从 1 开始

{

if (m%i == 0)

{

if (isFirstFactor == 0)

printf("%d", i);

printf(",");

isFirstFactor = 0;

}

}

printf("\n");

}

**9.** 亲密数**\_1**

#include <stdio.h>

#include <math.h>

int FactorSum(int x);

int main()

{

int m, n;

printf("Input m, n:\n");

scanf("%d,%d", &m, &n);

if (FactorSum(m)==n && FactorSum(n)==m) //若 m 和 n 是亲密数

{

printf("Yes!\n");

}

else

{

//若 m 和 n 不是亲密数

printf("No!\n");

}

return 0;

}

// 函数功能：返回 x 的所有因子之和

int FactorSum(int x)

{

int i;

int sum = 0;

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

{

if (x%i == 0)

{

sum = sum + i;

}

}

return sum;

}

**10.** 亲密数**\_2**

#include <stdio.h>

#include <math.h>

int FactorSum(int x);

int main()

{

int n, m, k, i;

printf("Input n:\n");

scanf("%d", &n);

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

{

m = FactorSum(i);//计算 i 的所有因子之和

k = FactorSum(m); //计算 m 的所有因子之和

if (i==k && i<m) //若 m 和 i 是亲密数

{

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

}

}

return 0;

}

// 函数功能：返回 x 的所有因子之和

int FactorSum(int x)

{

int i;

int sum = 0;

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

{

if (x%i == 0)

{

sum = sum + i;

}

}

return sum;

}