Chapter8-程序源代码与运行结果

8.1

(1) 10

(2) 0123

8.2

(1) [1] =0

[2] ++count

[3] count

(2) [1] int n

[2] f[i-1] + f[i-2]

(3) [1] 0

[2] a[n]>max

[3] n

[4] a[n]<min

[5] n

(4) [1] c[ROW][ROW]

[2] 0

[3] c[i][j] + a[i][k] + b[k][j]

[4] putchar(‘\n’)

[5] &a[i][j]

[6] &b[i][j]

[7] a, b, c

8.4

// Source Codes:

#include <stdio.h>

**int** countFail(**int** a[], **int** stuNum, **int** passScore) {

**int** i=0, count=0;

**for** (i=0 ; i<stuNum ; i++) **if** (a[i]<passScore) count++;

**return** count;

}

**int** main() {

**int** stuScore[40], passScore=0, stuNum=0, i=0;

printf("input student number > ");

scanf("%d", &stuNum);

printf("input pass score > ");

scanf("%d", &passScore);

printf("input scores > \n");

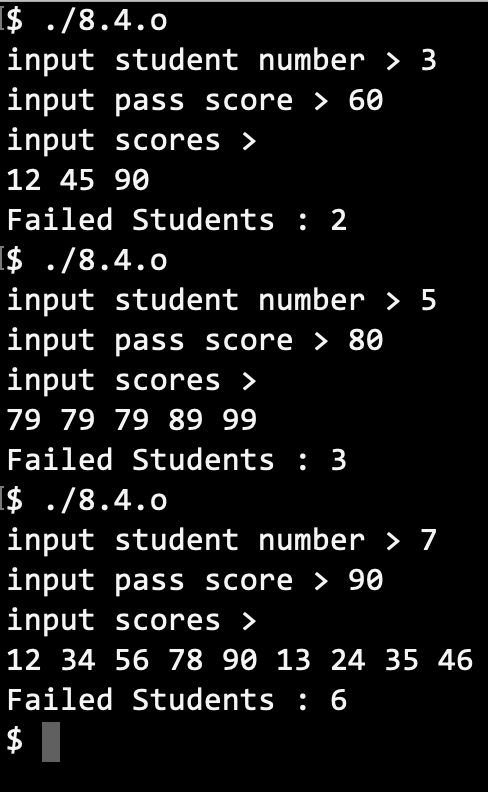
**for** (i=0 ; i<stuNum ; i++) scanf("%d", &stuScore[i]);

printf("Failed Students : %d\n", countFail(stuScore, stuNum, passScore));

**return** 0;

}

// Results:



8.10

// Source Codes:

#include <stdio.h>

**int** calculate\_main(**int**\* mat, **int** rank) {

**int** result=0, i=0, j=0;

**for** (i=0 ; i<rank\*rank ; i+=rank) result += mat[i+(j++)];

**return** result;

}

**int** calculate\_reverse(**int**\* mat, **int** rank) {

**int** result=0, i=0, j=rank-1;

**for** (i=0 ; i<rank\*rank ; i+=rank) result += mat[i+(j--)];

**return** result;

}

**int** main() {

**int** rank=0, i=0, j=0;

**int** cmain=0, crvs=0;

printf("input rank > ");

scanf("%d", &rank);

**const** **int** \_rank = rank;

**int** mat[\_rank\*\_rank];

**for** (i=0 ; i<\_rank\*\_rank ; i++) scanf("%d", &mat[i+j]);

cmain = calculate\_main(mat, \_rank);

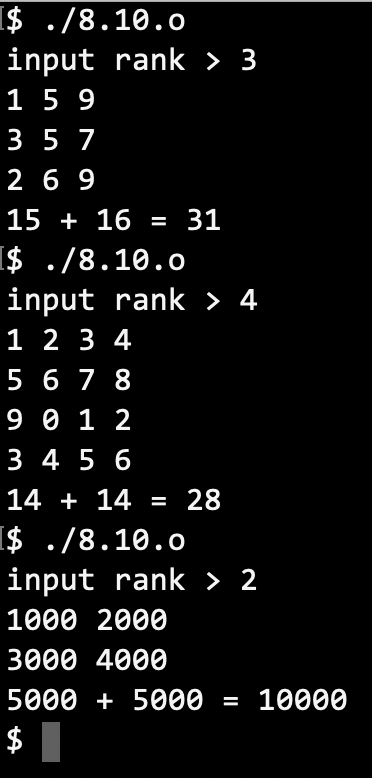
crvs = calculate\_reverse(mat, \_rank);

printf("%d + %d = %d\n", cmain, crvs, cmain+crvs);

**return** 0;

}

// Results:



8.12

// Source Codes:

#include <stdio.h>

#include <string.h>

**void** outputYangHui() {

**int** YangHui[7][7];

**int** i=0, j=0;

memset(YangHui, 0, **sizeof**(YangHui));

**for** (i=0 ; i<7 ; i++) {

YangHui[i][0] = 1;

YangHui[i][i] = 1;

}

**for** (i=2 ; i<7 ; i++) **for** (j=1 ; j<i ; j++)

YangHui[i][j] = YangHui[i-1][j-1] + YangHui[i-1][j];

**for** (i=0 ; i<7 ; i++) {

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

printf ("%-3d ", YangHui[i][j]);

putchar('\n');

}

}

**int** main() {

outputYangHui();

**return** 0;

}

// Result:

