Chapter8-程序源代码与运行结果

8.16

// Source Codes:

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

**void** selectionSort(**int**, **int**\*);

**int** find\_x\_pos(**int**\*, **int**, **int**\*, **int**);

**int** main() {

**int**\* array = malloc(10\***sizeof**(**int**));

**int** size = 10;

**int** i=0, x=0;

printf ("Input array : \n(Enter -1 to stop)\n");

**while**(1) {

**while** (!scanf ("%d", &array[i])) **while** (getchar()!='\n');

**if** (array[i]==-1) **break**;

i++;

**if** (i>=size-1) {

size += 10;

array = realloc(array, size);

}

}

selectionSort(i, array);

**do** {

printf("input x > ");

**while** (getchar()!='\n');

} **while** (!scanf ("%d", &x));

**int** x\_pos = find\_x\_pos(&size, i, array, x);

**int** k=0;

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

**if** (k==x\_pos) printf ("%d(x) ", array[k]);

**else** printf ("%d ", array[k]);

}

putchar('\n');

**return** 0;

}

**void** selectionSort(**int** len, **int**\* array) {

**int** i=0;

**for** (i=0 ; i<len-1 ; i++) {

**int** j=i;

**int** minPos = j;

**for** (j=i ; j<len ; j++) **if** (array[minPos]>array[j]) minPos = j;

**int** temp = array[i];

array[i] = array[minPos];

array[minPos] = temp;

}

}

**int** find\_x\_pos(**int**\* size, **int** len, **int**\* array, **int** x) {

**int** i=0;

**int** x\_pos = len;

**for** (i=0 ; i<len ; i++) **if** (x<array[i]) {

x\_pos = i;

**break**;

}

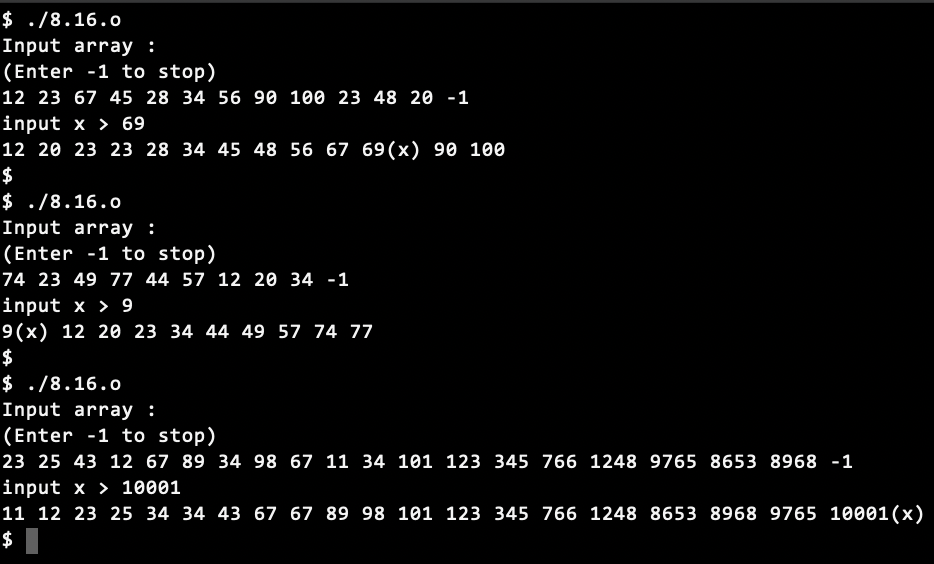
**for** (i=len ; i>x\_pos ; i--) array[i] = array[i-1];

array[x\_pos] = x;

**return** x\_pos;

}

// Results:



8.17

// Source Codes:

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#define N 40

**int** ReadScore(**int** score[]);

**void** DataSort(**int** score[], **int** n);

**void** PrintScore(**int** score[], **int** n);

**int** main() {

**int** score[N], n;

n = ReadScore(score);

DataSort(score, n);

printf ("Sorted scores : ");

PrintScore(score, n);

**return** 0;

}

**int** ReadScore(**int** score[]) {

**int** i=-1;

**do** {

i++;

printf ("input score > ");

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

} **while** (score[i]>=0);

**return** i;

}

**void** DataSort(**int** score[], **int** n) {

**bool** moved = **false**;

**int** i=0;

**do** {

moved = **false**;

**for** (i=0 ; i<n-1 ; i++) **if** (score[i]<score[i+1]) {

**int** swap=score[i];

score[i] = score[i+1];

score[i+1] = swap;

moved = **true**;

}

} **while**(moved);

}

**void** PrintScore(**int** score[], **int** n) {

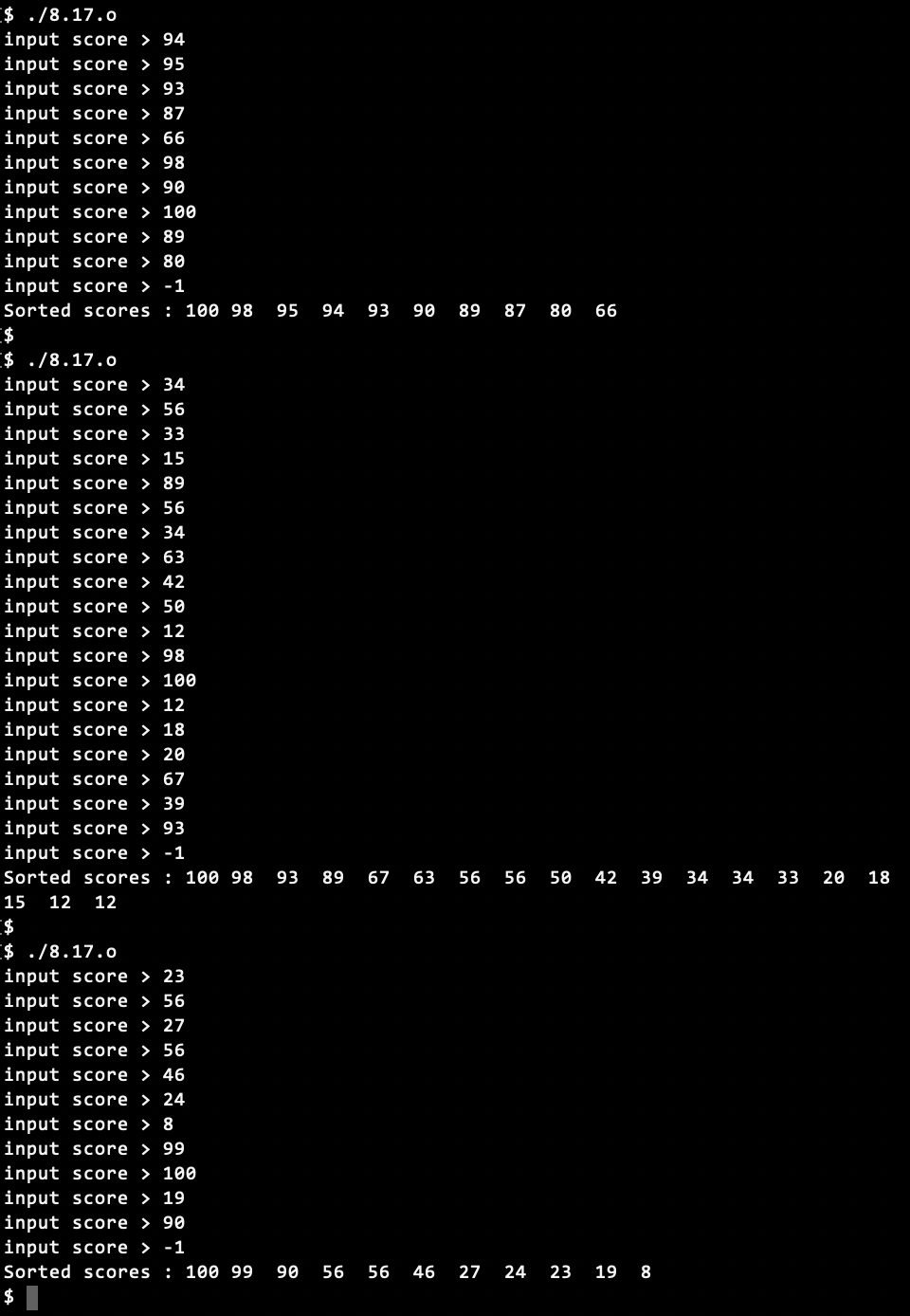
**int** i;

**for** (i=0 ; i<n ; i++) printf("%-4d", score[i]);

putchar('\n');

}

// Results:



8.18

// Source Codes:

#include <stdio.h>

#include <string.h>

**const** **int** DIGITS = 50;

**void** refresh\_m(**int** massiveNumber[DIGITS]); /\* "refresh" conflict with system function \*/

**void** output(**int** massiveNumber[DIGITS]);

**int** main() {

**int** massiveNumber[DIGITS];

memset(massiveNumber, -1, DIGITS\***sizeof**(**int**));

massiveNumber[0] = 1;

**int** order = 0;

**do** {

fflush(stdin);

printf ("Enter a number to be calculated > ");

} **while**(!scanf("%d", &order));

**int** i=0;

**for** (i=1 ; i<=order ; i++) {

**int** j=0;

**for** (j=0 ; massiveNumber[j]!=-1 ; j++) massiveNumber[j] \*= i;

refresh\_m(massiveNumber);

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

output(massiveNumber);

putchar('\n');

}

}

**void** refresh\_m(**int** massiveNumber[DIGITS]) {

**int** i=0;

**for** (i=0 ; i<DIGITS && massiveNumber[i]!=-1 ; i++) {

**while** (massiveNumber[i]>=10) {

**if** (massiveNumber[i+1]==-1) massiveNumber[i+1] = 0;

massiveNumber[i] -= 10;

massiveNumber[i+1] += 1;

}

}

}

**void** output(**int** massiveNumber[DIGITS]) {

**int** i=0;

**for** (i=DIGITS-1 ; i>=0 ; i--) **if** (massiveNumber[i]!=-1) printf("%d", massiveNumber[i]);

}

// Result:

