

目录

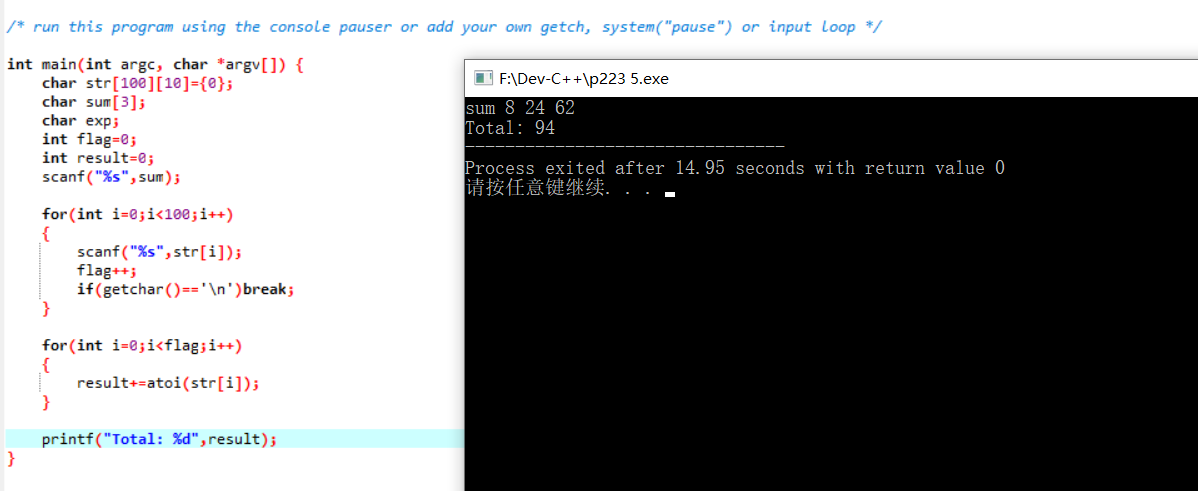
[（1）教材第13章编程题第5题 2](#_Toc26996394)

[（2）教材第16章编程题第5题 2](#_Toc26996395)

[（3）教材第17章编程题第5题 2](#_Toc26996396)

（1）教材第13章编程题第5题：编写程序实现对输入命令行参数求和。

说明：本题加深对命令行参数的理解、考查知识的综合应用。

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

char str[100][10]={0};

char sum[3];

char exp;

int flag=0;

int result=0;

scanf("%s",sum);

for(int i=0;i<100;i++)

{

scanf("%s",str[i]);

flag++;

if(getchar()=='\n')break;

}

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

{

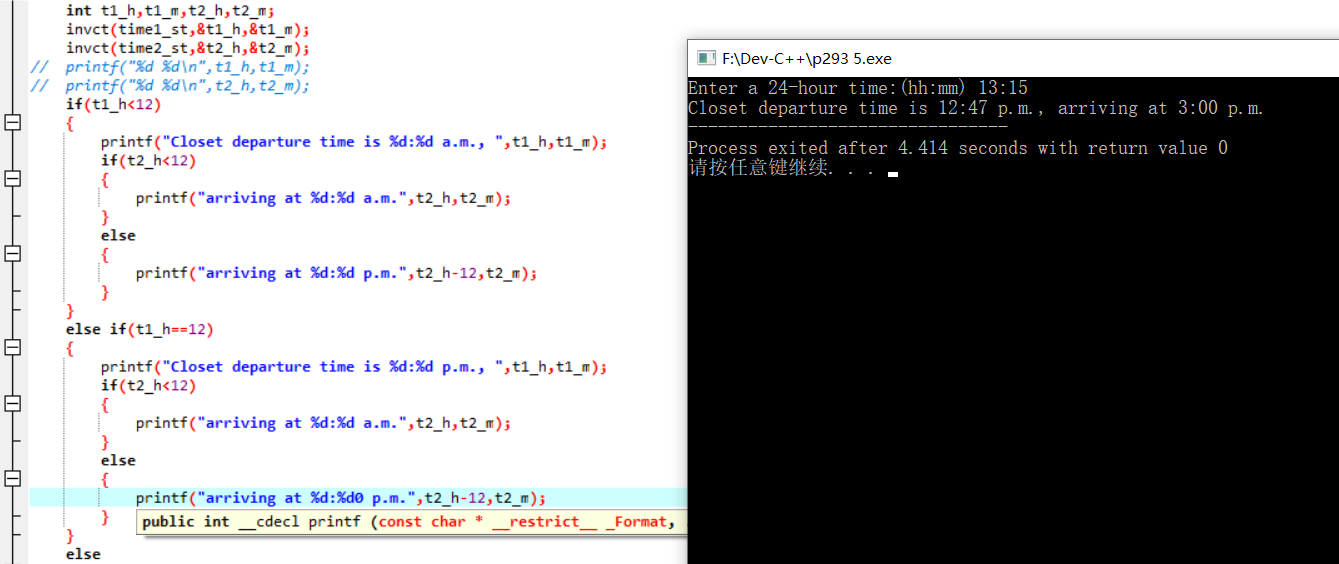
result+=atoi(str[i]);

}

printf("Total: %d",result);

}

（2）教材第16章编程题第5题：用结构变量解决搜索离用户输入时间最近的航班信息，并输出。

说明：本题考查结构与数组的结合运用，有助于理解结构。首先，定义一个存储起飞、到达时间的结构，再定义一个该类型的数组，循环搜索即可。#include <stdio.h>

#include <stdlib.h>

#include <math.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int ct(int a,int b)

{

return a\*60+b;

}

void invct(int time,int\* h,int\* m)

{

\*h=time/60;

\*m=time-(\*h)\*60;

}

int main(int argc, char \*argv[]) {

struct TIME{

int start\_time;

int arrival\_time;

}time[8]={ {ct(8,0),ct(10,16)}, {ct(9,43),ct(11,52)},{ct(11,19),ct(13,31)},{ct(12,47),ct(15,0)},{ct(14,0),ct(16,8)},{ct(15,45),ct(17,55)},{ct(19,0),ct(21,20)},{ct(21,45),ct(23,58)}};

printf("Enter a 24-hour time:(hh:mm) ");

int time1,time2,timeg;

scanf("%d:%d",&time1,&time2);

timeg=ct(time1,time2);

// printf("%d\n",timeg);

int time1\_st;

int time2\_st;

int min=fabs(timeg-time[0].start\_time);

for(int i=0;i<8;i++)

{

if(min > fabs(timeg-time[i].start\_time))

{

min=fabs(timeg-time[i].start\_time);

time1\_st=time[i].start\_time;

time2\_st=time[i].arrival\_time;

}

}

// printf("%d %d\n",time1\_st,time2\_st);

int t1\_h,t1\_m,t2\_h,t2\_m;

invct(time1\_st,&t1\_h,&t1\_m);

invct(time2\_st,&t2\_h,&t2\_m);

// printf("%d %d\n",t1\_h,t1\_m);

// printf("%d %d\n",t2\_h,t2\_m);

if(t1\_h<12)

{

printf("Closet departure time is %d:%d a.m., ",t1\_h,t1\_m);

if(t2\_h<12)

{

printf("arriving at %d:%d a.m.",t2\_h,t2\_m);

}

else

{

printf("arriving at %d:%d p.m.",t2\_h-12,t2\_m);

}

}

else if(t1\_h==12)

{

printf("Closet departure time is %d:%d p.m., ",t1\_h,t1\_m);

if(t2\_h<12)

{

printf("arriving at %d:%d a.m.",t2\_h,t2\_m);

}

else

{

printf("arriving at %d:%d0 p.m.",t2\_h-12,t2\_m);

}

}

else

{

printf("Closet departure time is %d:%d p.m., ",t1\_h-12,t1\_m);

if(t2\_h<12)

{

printf("arriving at %d:%d a.m.",t2\_h,t2\_m);

}

else

{

printf("arriving at %d:%d p.m.",t2\_h-12,t2\_m);

}

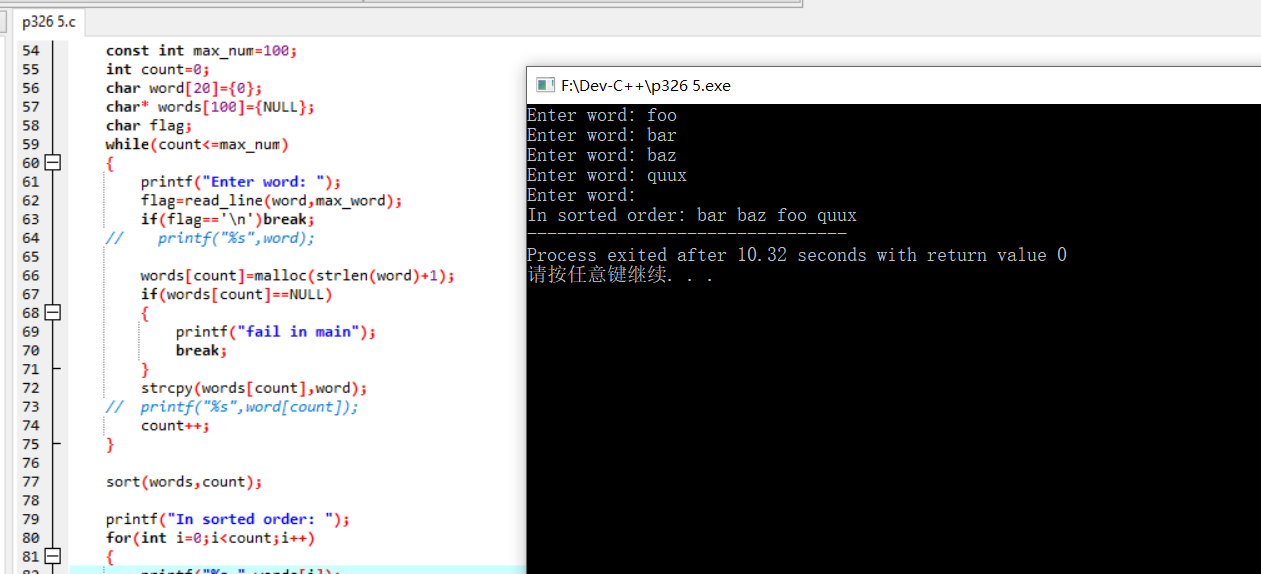
}

return 0;

}

（3）教材第17章编程题第5题：用指针数组存储键入的字符串，对数组排序。

说明：本题应用动态分配内存、数组排序，考查综合能力。

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

//first:read\_line word to a array[20].............bingo

//second:save the word to a \*[100]

//third:sort the array

//fourth:print the array

char read\_line(char word[],int max\_word)

{

char y;

if((y=getchar())=='\n')

{

return y;

}

word[0]=y;

for(int i=1;(y=getchar())!='\n' && i<max\_word;i++)

{

word[i]=y;

}

return 'a';

}

void sort(char\* words[],int num)

{

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

{

for(int j=i;j<num;j++)

{

if(strcmp(words[i],words[j])>0)

{

swap(words[i],words[j]);

}

}

}

}

void swap(char\* a,char\*b)

{

char\* c=malloc(strlen(a)+1);

if(c==NULL)

{

printf("fail in swap");

return NULL;

}

strcpy(c,a);

strcpy(a,b);

strcpy(b,c);

}

int main(int argc, char \*argv[]) {

const int max\_word=20;

const int max\_num=100;

int count=0;

char word[20]={0};

char\* words[100]={NULL};

char flag;

while(count<=max\_num)

{

printf("Enter word: ");

flag=read\_line(word,max\_word);

if(flag=='\n')break;

// printf("%s",word);

words[count]=malloc(strlen(word)+1);

if(words[count]==NULL)

{

printf("fail in main");

break;

}

strcpy(words[count],word);

// printf("%s",word[count]);

count++;

}

sort(words,count);

printf("In sorted order: ");

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

{

printf("%s ",words[i]);

}

return 0;

}