Week\_2

1.

#include <stdio.h>

#define MAX\_LEN 10

double get\_result(const double \*,int);

int main(void){

double arr[MAX\_LEN];

int i;

puts("input data(num:10):");

for(i = 0;i < MAX\_LEN;i ++)

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

printf("result:%lf\n",get\_result(arr,MAX\_LEN));

return 0;

}

double get\_result(const double \* ap,int len){

double result = 1;

int i;

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

result \*= ap[i];

return result;

}

2.

#include <stdio.h>

#define MAX\_LEN 6

void sort(int \*,int);

int main(void){

int arr[MAX\_LEN];

int i;

puts("input data(num:5):");

for(i = 0;i < MAX\_LEN - 1;i ++)

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

sort(arr,MAX\_LEN - 1);

puts("insert num:");

scanf("%d",&arr[MAX\_LEN - 1]);

sort(arr,MAX\_LEN);

return 0;

}

void sort(int \* ip,int len){

int i;

int j;

int temp;

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

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

if(ip[i] > ip[j]){

temp = ip[i];

ip[i] = ip[j];

ip[j] = temp;

}

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

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

putchar('\n');

}

3.

#include <stdio.h>

#define LEN 10

static int a[LEN] = {132,443,17,325,128,55,589,250,666,999};

void sort(int \*,int);

int main(void){

int new[LEN];

int i;

int index = 0;

printf("Deleted:");

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

if(a[i] % 5 == 0)

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

else{

new[index] = a[i];

index ++;//following index is len

}

putchar('\n');

sort(new,index);

return 0;

}

void sort(int \* ip,int len){

int i;

int j;

int temp;

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

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

if(ip[i] > ip[j]){

temp = ip[i];

ip[i] = ip[j];

ip[j] = temp;

}

printf("Sorted:");

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

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

putchar('\n');

}

4.

#include <stdio.h>

#define MAX\_LEN 10

int main(void) {

int arr[MAX\_LEN],result[MAX\_LEN];

int start,end;

int i;

int temp = 0;

int lens;

int max;

puts("input the list(num:10):");

for(i = 0;i < MAX\_LEN;i ++)

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

max = arr[0];

for(start = 0;start < MAX\_LEN;start ++)

for(end = MAX\_LEN - 1;end >= start;end --){

for(i = start;i < end;i ++){

temp += arr[i];

}

if (temp > max){

max = temp;

lens = 0;

for(i = start;i < end;i ++){

result[i] = arr[i];

lens ++;

}

}

temp = 0;

}

printf("MAX:%d\n",max);

printf("List:");

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

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

putchar('\n');

return 0;

}

5.

#include <stdio.h>

#define MAX\_LEN (10+6+1)

//+3.4E+38

char str[MAX\_LEN];

char \* myFloatToString(float);

int pow\_ten(int);

int main(void) {

float f;

printf("input float:");

scanf("%f",&f);

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

printf("%s\n",myFloatToString(f));

return 0;

}

int pow\_ten(int num){

int temp = 1;

for(;num > 0;num --)

temp \*= 10;

return temp;

}

char \* myFloatToString(float f){

int i;

int index = 0;

int r;

int int\_part = (int)f;

float float\_part = f - int\_part;

\_Bool flag = 0;

for(i = 10;i >= 0;i --){

r = int\_part / pow\_ten(i);

if(r != 0)

flag = 1;

if(flag){

int\_part -= r \* pow\_ten(i);

str[index] = '0'+r;

index ++;

}

}

str[index] = '.';

index ++;

int\_part = (int) (float\_part \* pow\_ten(6));

for(i = 5;i >= 0;i --){

r = int\_part / pow\_ten(i);

int\_part -= r \* pow\_ten(i);

str[index] = '0'+ r;

index ++;

}

str[index] = '\0';

return str;

}

6.

/\*--------------------------------------------------------------------------

拼不出的单词

Copyright (c) 2017 HDU\_Microchip\_Lab Co., Ltd.

2017-08-24

--------------------------------------------------------------------------\*/

#include <stdio.h>

#include <string.h>

#define LENGTH\_OF\_LIB 3 //词库中词的数量

const char Library[LENGTH\_OF\_LIB][10] = { "microchip","microwave","microsoft" }; /\* use string.h(strlen) must make sure the last ele is '\0' 9 -> 10 \*/ //词库

/\* 20 -> 10 \*/

int searchWord(char Input\_Word[], const char Library[][10]); //搜索单词函数

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

{

int i = searchWord("microwave", Library);

if (-1 != i)

{

printf("%s\n", Library[i]);

}

else

{

printf("Word not found!\n");

}

getchar();/\*getch -> getchar , getch is not in head file stdio.h \*/

return 0;

}

/\* 20 -> 9 \*/

int searchWord(char Input\_Word[], const char Library[][10])

{

int Num\_Of\_Word; //词库索引

int CNT\_Same\_Char = 0; //相同字符计数

int Length\_Of\_Word = strlen(Input\_Word); /\* strlen recall value is an interge,drop (int) \*/ //当前单词长度

char Index[20] = { 0 }; //用于输入字符串操作

int i, j = 0;

for (i = 0; i<Length\_Of\_Word; i++) //将输入字符串保存到操作数组中

{

Index[i] = Input\_Word[i];

}

for (Num\_Of\_Word = 0; Num\_Of\_Word<LENGTH\_OF\_LIB; Num\_Of\_Word++) //遍历词库中所有单词

{

for (; Num\_Of\_Word<LENGTH\_OF\_LIB && Length\_Of\_Word != strlen(Library[Num\_Of\_Word]); Num\_Of\_Word++); //如果字符串长度不同，直接进入下一次查找

if (Num\_Of\_Word >= LENGTH\_OF\_LIB)

{

return -1;

}

for (i = 0; i<Length\_Of\_Word; i++) //遍历一个单词中的所有字符

{ /\*Index[j] i -> j \*/

for (j = 0; j<Length\_Of\_Word && Index[j] != Library[Num\_Of\_Word][i]; j++); //j不停计数，直到操作数组的当前字符与库中当前单词中的某一个字符相同

if (j < Length\_Of\_Word)

{

for (; j<Length\_Of\_Word; j++) //将相同的字符从操作数组中移除

{

Index[j] = Index[j + 1];

}

CNT\_Same\_Char++; //相同字符计数增加

}

if (CNT\_Same\_Char == Length\_Of\_Word) //如果所有字符都与当前字库中的单词相同

{

return Num\_Of\_Word;

}

}

}

return -1; //如果匹配结束仍未找到相同单词，返回-1表示未找到

}