

算法题

```
#include <stdlib.h>
#include <stdio.h>
void main() {
    FILE *in, *out;
    char ch, filename1[100], filename2[100];
    scanf("%s", filename1);
    scanf("%s", filename2);
    if ((in = fopen(filename1, "r")) == NULL)
    {
        printf("无法打开此文件\n");
        exit(0);
    }

    if ((out = fopen(filename2, "w")) == NULL)
    {
        printf("无法打开此文件\n");
        exit(0);
    }
    ch = fgetc(in);
    while (!feof(in))
    {
        fputc(ch, out);
        ch = fgetc(in);
    }
    fclose(in);
    fclose(out);
}

```c
```

```
#include <stdio.h>

// 字符串连接
void strCat(char *p1, char *p2)
{
 while (*p1)
 {
 p1++;
 }
}
```

```
while (*p2)
{
 *p1++ = *p2++;
}
*p1 = '\0';
}
```

// 字符串比较

```
int strCmp(char *p1, char *p2)
{
 while (*p1 && *p2 && (*p1 == *p2))
 {
 p1++;
 p2++;
 }
 return *p1 - *p2;
}
```

// 字符串复制

```
void strCpy(char *p1, char *p2)
{
 while (*p2)
 {
 *p1++ = *p2++;
 }
 *p1 = '\0';
}
```

// 折半查找

```
void binarySearch(int arr[], int len, int key)
{
 int low = 0, high = len - 1, mid;
 while (low <= high)
 {
 mid = (low + high) / 2;
 if (arr[mid] > key)
 {
 high = mid - 1;
 }
 else if (arr[mid] < key)
 {
 low = mid + 1;
 }
 }
}
```

```

 else
 {
 printf("关键字为%d的下标为%d\n", key, mid);
 return;
 }
}
printf("关键字%d未找到\n", key);
// printf("关键字为%d的下标为%d\n", key, mid);
// printf("关键字%d未找到\n", key);
}

```

// 插入排序

```

void insertSort(int arr[], int len)
{
 for (int i = 1; i < len; i++)
 {
 int j = i - 1;
 int curr = arr[i];
 while (j >= 0 && arr[j] > curr)
 {
 arr[j + 1] = arr[j];
 j--;
 }
 arr[j + 1] = curr;
 }
}

```

// 选择排序

```

void selectSort(int arr[], int len)
{
 for (int i = 0; i < len; i++)
 {
 int curr = i;
 for (int j = i + 1; j < len; j++)
 {
 if (arr[j] < arr[curr])
 {
 curr = j;
 }
 }
 if (curr != i)
 {
 int num = arr[i];

```

```

 arr[i] = arr[curr];
 arr[curr] = num;
 }
}
}

```

// 冒泡排序

```

void bubbleSort(int arr[], int len)
{
 for (int i = 0; i < len; i++)
 {
 for (int j = 0; j < len - i; j++)
 {
 if (arr[j] > arr[j + 1])
 {
 int num = arr[j];
 arr[j] = arr[j + 1];
 arr[j + 1] = num;
 }
 }
 }
}

```

```

void main()
{
 void insertSort(int arr[], int len);
 void selectSort(int arr[], int len);
 void bubbleSort(int arr[], int len);
 int arr[] = {4, 2, 1, 5, 10, 6};
 int len = 6;
 int arr1[] = {1, 3, 7, 9, 11, 22, 63};

 // insertSort(arr, len);
 selectSort(arr, len);
 // bubbleSort(arr, len);

 // binarySearch(arr1, 7, 22);
 // 遍历
 for (int i = 0; i < len; i++)
 {
 printf("%d ", arr[i]);
 }
}

```

```
char s1[20];
char s2[20] = "i love ";
char s3[20] = "hui cai lin";
char *p1 = s1, *p2 = s2;
char *p3 = s3;
// strcpy(s1, s2); // 传递地址
// printf("%d", strcmp(p2, p3)); // 传递地址
// printf("%d", strcmp(p2, p3));
// strcat(s2, s3);
// printf("%s", p2);
// puts(s1);
}
```