2020 期末考

单选题

CAACB
CBDBC
ADAAB
CDCAC
DBCBB

多选题

AD
BCD
ABE
ABCE
ACDE
CDE

填空

1. `n % 100 / 10` 或 `n / 10 % 10`
2. `s[j] = s[i]`
3. `j++`
4. `abclo`
5. `abc`
6. `a[i] == x`
7. `i`
8. `-1`
9. `(double)t / (2 * i + 1)`
10. `s`

阅读程序

1. 10
2. 13
3. bbbccc
4. acekills
5. k=14

编程应用

1

```c
int fact(int n) {
    int product = 1;
    for (int i = 1; i <= n; i++)
        product *= i;
    return product;
}

int *coef(int n) {
    if (n < 0)
        return NULL;
    int *result = (int *)malloc(sizeof (int) * (n + 1));
    for (int i = 0; i <= n; i++)
        result[i] = fact(n) / (fact(i) * fact(n - i));
    return result;
}
```

2

```c
int readdata(char *fname, int d[]) {
    FILE *fp = fopen(fname, "w");
    if (fp == NULL)
        return 0;
    int n = 0;
    for (; n < 100 && fscanf(fp, "%d", d + n) == 1; n++);
    fclose(fp);
    return n;
}
int median(int d[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - 1 - i; j++) {
            if (d[j] > d[j + 1]) {
                int tmp = d[j];
                d[j] = d[j + 1];
                d[j + 1] = tmp;
            }
        }
    }
    return n % 2 ? d[n / 2] : (d[n / 2 - 1] + d[n / 2]) / 2;
 }
```

3

```
while (slen = strlen(str)) {
    struct dict *new_node = malloc(sizeof(struct dict));
    new_node->entry = malloc(sizeof (char) * (slen + 1));
    new_node->next = NULL;
    strcpy(new_node->entry, str);
    head = sortdict(head, new_node);
    gets(str);
}

struct dict *sortdict(struct dict *head, struct dict *ps) {
    struct dict *previous = NULL;
    struct dict *current = head;
    for (; current != NULL && strcmp(current->entry, ps->entry) < 0;)
{
        previous = current;
        current = current->next;
    }
    if (previous == NULL)
        return ps;
    ps->next = current;
    previous->next = ps;
```

```c
    return head;
}

void printdict(struct dict *head) {
    for (; head != NULL; head = head->next) {
        printf("%s\n", head->entry);
    }
    /* 严格来讲，这里还要释放整个链表的内存，因为 printdict 函数是最后调用的函数
    for (; head != NULL;) {
        struct dict *tmp = head;
        head = head->next;
        free(tmp);
    }
    */
}
```