

cookbook

Q (A)

参考的答案错误源代码如下:

```
1  #include <stdio.h>
2  #include <math.h>
3  #define n 10000
4
5  double f1(double x);
6  double f2(double x);
7  double definiteIntegral(double (*f)(double), double a, double b);
8
9  int main()
10 {
11     double a1, b1,a2, b2;
12     scanf("%lf %lf", &a1, &b1);
13     //getchar();
14     scanf("%lf %lf", &a2, &b2);
15
16
17     //f1(x)
18     double result1 = definiteIntegral(f1, a1, b1);
19     printf("%.4lf\n", result1);
20
21     //f2(x)
22     double result2 = definiteIntegral(f2, a2, b2);
23     printf("%lf", result2);
24
25     return 0;
26 }
27
28 double f1(double x) {
29     return x * x + 2 * x + 9;
30 }
31
32 double f2(double x) {
33     return sin(x / 2) * sin(x / 2);
34 }
35
36 double definiteIntegral(double (*f)(double), double a, double b)
37 {
38     double sum = 0.0;
39     double width = (b - a) / n;
40     for (int i = 0; i < n; i++) {
41         double x = a + i * width;
42         sum += f(x)* width;
43     }
44     return sum;
45 }
```

运行结果：

```
C:\WINDOWS\system32\cmd. X + v
1.0 2.0
0.0 1.57
14.333083
0.284961
Press any key to continue . . .
```

wa的原因：

结果的精度略高于正确结果。大概是因为printf的精度和cpp中的cout的精度不一样，这道题我印象中原本是给cpp做的，不用特殊考虑处理精度。但是现在拿过来给c做，需要人为控制一下精度，在主函数中做如下修改就能ac了。

修正：

```
1  int main()
2  {
3      double a1, b1, a2, b2;
4      scanf("%lf %lf", &a1, &b1);
5      //getchar();
6      scanf("%lf %lf", &a2, &b2);
7
8      //f1(x)
9      double result1 = definiteIntegral(f1, a1, b1);
10     printf("%.4lf\n", result1);
11
12     //f2(x)
13     double result2 = definiteIntegral(f2, a2, b2);
14     printf("%lf", result2);
15
16     return 0;
17 }
```

其实是题目的问题，不用太纠结，知道怎么控制精度就可以了qwq

Q (G)

参考的运行异常源代码如下：

```
1  #include <stdio.h>
2
3  int circle(int num);
4
5  int main()
6  {
7      int n;
8      scanf("%d ", &n);
9      while(n--){
10         int num, t;
```

```

11     scanf("%d ", &num);
12     t=circle(num);
13     printf("%d\n", t);
14 }
15 return 0;
16 }
17
18
19 int circle(int num)
20 {
21     int a=0, b=0, count=1;
22     int arr[4], temp;
23     int new_num;
24     arr[0] = num / 1000;
25     arr[1] = (num / 100) % 10;
26     arr[2] = (num / 10) % 10;
27     arr[3] = num % 10;
28     for(int i = 0; i < 4; i++){
29         for(int j = 0; j < 4-i; j++){
30             if(arr[j] > arr[j+1]){
31                 temp = arr[j];
32                 arr[j] = arr[j+1];
33                 arr[j+1] = temp;
34             }
35         }
36     }
37     a=arr[0]*1000+arr[1]*100+arr[2]*10+arr[3];
38     b=arr[3]*1000+arr[2]*100+arr[1]*10+arr[0];
39     new_num=a-b;
40
41     while(num != new_num){
42         arr[0] = new_num / 1000;
43         arr[1] = (new_num / 100) % 10;
44         arr[2] = (new_num / 10) % 10;
45         arr[3] = new_num % 10;
46         for(int i = 0; i < 4; i++){
47             for(int j = 0; j < 4-i; j++){
48                 if(arr[j] > arr[j+1]){
49                     temp = arr[j];
50                     arr[j] = arr[j+1];
51                     arr[j+1] = temp;
52                 }
53             }
54         }
55         a=arr[0]*1000+arr[1]*100+arr[2]*10+arr[3];
56         b=arr[3]*1000+arr[2]*100+arr[1]*10+arr[0];
57         new_num=a-b;
58         count++;
59     }
60     return count;
61 }

```

存在的问题:

- 1 | `scanf("%d ",&n);scanf("%d ",&num);`

`scanf` 在遇到空白字符时, 会跳过所有连续的空白字符, 直到遇到第一个非空白字符为止。

- 数组越界

```
1  for(int i = 0; i < 4; i++){
2      for(int j = 0; j < 4-i; j++){
3          if(arr[j] > arr[j+1]){
4              temp = arr[j];
5              arr[j] = arr[j+1];
6              arr[j+1] = temp;
7          }
8      }
9  }
```

`arr` 数组开的4, 内层循环到j=4的时候会进行arr[4]和arr[5]的比较, 这会导致数组越界的问题
需要去规范一下冒泡排序的写法。

- 题意理解错误

比较的是更新完的数字和上次更新前的数字, 而不是和最开始的数字。

ex: 从1234出发, 依次可以得到4321-1234=3087、8730-378=8352、8532-2358=6174, 又回到了它自己。这里的它自己是6174。

- `new_number`计算错误

数组排序之后大的数字在后面, `new_number`应为b-a

修正:

main函数:

```
1  int main()
2  {
3      int n;
4      scanf("%d",&n);
5      while(n--){
6          int num,t;
7          scanf("%d",&num);
8          t=circle(num);
9          printf("%d\n",t);
10     }
11     return 0;
12 }
```

circle函数:

```
1  int circle(int num)
2  {
3      int a=0,b=0,count=1;
4      int arr[4],temp;
```

```

5     int new_num;
6     arr[0] = num / 1000;
7     arr[1] = (num / 100) % 10;
8     arr[2] = (num / 10) % 10;
9     arr[3] = num % 10;
10    for(int i = 0; i < 4; i++){
11        for(int j = 0; j < 3-i; j++){
12            if(arr[j] > arr[j+1]){
13                temp = arr[j];
14                arr[j] = arr[j+1];
15                arr[j+1] = temp;
16            }
17        }
18    }
19    a=arr[0]*1000+arr[1]*100+arr[2]*10+arr[3];
20    b=arr[3]*1000+arr[2]*100+arr[1]*10+arr[0];
21    new_num=b-a;
22
23    while(num != new_num){
24        num = new_num;
25        arr[0] = new_num / 1000;
26        arr[1] = (new_num / 100) % 10;
27        arr[2] = (new_num / 10) % 10;
28        arr[3] = new_num % 10;
29        for(int i = 0; i < 4; i++){
30            for(int j = 0; j < 3-i; j++){
31                if(arr[j] > arr[j+1]){
32                    temp = arr[j];
33                    arr[j] = arr[j+1];
34                    arr[j+1] = temp;
35                }
36            }
37        }
38        a=arr[0]*1000+arr[1]*100+arr[2]*10+arr[3];
39        b=arr[3]*1000+arr[2]*100+arr[1]*10+arr[0];
40        new_num=b-a;
41        count++;
42    }
43    return count;
44 }

```

Q(H)

参考的答案错误的源码如下:

```

1  #include<stdio.h>
2  #include<string.h>
3
4  int word(char a);
5  void MaxLenword(char s[]);

```

```
6
7 int main()
8 {
9     int t;
10    scanf("%d",&t);
11    while(t--){
12        char s[1000];
13        gets(s);
14        MaxLenWord(s);
15    }
16    return 0;
17 }
18
19
20 int word(char a){
21     if ((a <= 'z' && a>='a') || (a <= 'Z' && a>='A')) {
22         return 1;
23     }
24     else
25     {
26         return 0;
27     }
28 }
29
30 void MaxLenWord(char s[]) {
31     int len=0;//长度
32     int maxlen=0;//最长的
33     int sign=0;
34     for (int i = 0; i<=strlen(s); i++)
35     {
36         if (word(s[i])) {
37             len++;
38         }
39         else
40         {
41             if (len>maxlen) {
42                 maxlen = len;
43                 sign = i -maxlen;
44                 len = 0;
45             }
46             len = 0;
47         }
48     }
49     for (int k = 0; k <=maxlen; k++)
50     {
51         printf("%c", s[sign + k]);
52     }
53 }
```

运行结果：

```
C:\WINDOWS\system32\cmd. X + v
2
It was rush hour and I was dashing to a train in New York City Grand
dashing
Press any key to continue . . . |
```

存在问题：

- `get()` 是一个已经弃用的函数，`scanf`会残留一个`\n`,`get()`第一次会直接把这个换行符给读进去
- `MaxLenWord`函数设计存在一些问题，比如没有考虑一句话中有多个长度最长的单词

解决办法：

- `scanf`问题：
 - 添加一个`getchar()`读掉换行符
- 重构`MaxLenWord`函数

完整的代码：

```
1  #include <stdio.h>
2  #include <string.h>
3
4  void MaxLenWord(char s[]) {
5      int len = strlen(s);
6      int maxLen = 0;
7      int currentLen = 0;
8
9      for(int i = 0; i <= len; i++) {
10         if(s[i] == ' ' || s[i] == '\0') {
11             if(currentLen > maxLen) {
12                 maxLen = currentLen;
13             }
14             currentLen = 0;
15         } else {
16             currentLen++;
17         }
18     }
19
20     currentLen = 0;
21     int isFirst = 1;
22
23     for(int i = 0; i <= len; i++) {
24         if(s[i] == ' ' || s[i] == '\0') {
25             if(currentLen == maxLen) {
26                 if(!isFirst) {
27                     printf(" ");
28                 }
29                 for(int j = i - currentLen; j < i; j++) {
```

```
30         printf("%c", s[j]);
31     }
32     isFirst = 0;
33 }
34     currentLen = 0;
35 } else {
36     currentLen++;
37 }
38 }
39 printf("\n");
40 }
41
42 int main() {
43     int t;
44     char s[1000];
45
46     scanf("%d", &t);
47     getchar();
48
49     while(t--) {
50         //fgets(s, sizeof(s), stdin);
51         gets(s);
52         MaxLenWord(s);
53     }
54
55     return 0;
56 }
```