

```
1  #include <stdio.h>
2
3  unsigned long sums(int, int);
4
5
6  int main(void)
7  {
8      int digit, level;
9      unsigned long ssum=0;
10
11      // printf("请输入基数和项数:\n");
12      scanf("%d%d", &digit, &level);
13      ssum = sums(digit, level);
14      printf("%lu", ssum);
15
16      return 0;
17  }
18
19
20 unsigned long sums(int digit, int level)
21 {
22     unsigned long sum=0, item=digit;
23
24     while (level--) {
25         sum += item;
26         item = digit + 10*item;
27     }
28
29     return sum;
30 }
31
```

```
1  #include <stdio.h>
2
3
4  void reduction(int a, int b)
5  {
6      int m, n, r;
7
8      if (a<=0 || b<=0) {
9          printf("分子分母必须为正整数");
10         return;
11     }
12
13     m = a;
14     n = b;
15     while (r = m%n) {
16         m = n;
17         n = r;
18     }
19
20     if (n==1)
21         printf("%d/%d为最简分数", a, b);
22     else
23         printf("%d/%d=%d/%d", a, b, a/n, b/n);
24 }
25
26
27 int main()
28 {
29     int a, b;
30
31     scanf("%d%d", &a, &b);
32     reduction(a, b);
33
34     return 0;
35 }
36
```

```
1  #include <stdio.h>
2
3  struct complex {
4      double real;
5      double imag;
6  };
7
8
9  struct complex complexMul(struct complex c1, struct complex c2)
10 {
11     struct complex p;
12
13     p.real = c1.real*c2.real - c1.imag*c2.imag;
14     p.imag = c1.real*c2.imag + c1.imag*c2.real;
15
16     return p;
17 };
18
19
20 int main()
21 {
22     struct complex c1, c2, p;
23
24     scanf("%lf%lfi", &c1.real, &c1.imag);
25     scanf("%lf%lfi", &c2.real, &c2.imag);
26     p = complexMul(c1, c2);
27     printf("%g%+gi", p.real, p.imag);
28
29     return 0;
30 }
31
```

```
1  #include <stdio.h>
2  #define ELE_NUM 10
3
4  float findMaxEle(float [], int);
5
6
7  int main()
8  {
9      int i;
10     float ele[ELE_NUM], maxEle=0.0;
11
12     for (i=0; i<ELE_NUM; i++) {
13         scanf("%f", &ele[i]);
14     }
15     maxEle = findMaxEle(ele, ELE_NUM);
16     printf("%f\n", maxEle);
17
18     return 0;
19 }
20
21
22 float findMaxEle(float a[], int n)
23 {
24     int i;
25     float max=a[0];
26
27     for (i=1; i<n; i++)
28         if (a[i]>max)
29             max = a[i];
30
31     return max;
32 }
33
```

```
1  #include <stdio.h>
2
3
4  long func_fac(int n)
5  {
6      long y=1;
7
8      for (int i=2; i<=n; i++)
9          y *= i;
10
11     return y;
12 }
13
14
15 double func_pow(double x, int n)
16 {
17     double y=1.0;
18
19     for (int i=0; i<n; i++)
20         y *= x;
21
22     return y;
23 }
24
25
26 double func_sum(double x, int n)
27 {
28     double y=0.0;
29
30     for (int i=0; i<=n; i++)
31         y += func_pow(x, i) / func_fac(i);
32
33     return y;
34 }
35
36
37 int main()
38 {
39     double x;
40     int    n;
41
42     scanf("%lf%d", &x, &n);
43     printf("%f", func_sum(x, n));
44
45     return 0;
46 }
47
```

```

1  #include <stdio.h>
2
3
4  void getString(char str[])
5  {
6      int i;
7      char c;
8
9      for (i=0; (c=getchar())!='\n'; i++)
10         str[i] = c;
11     str[i] = '\0';
12 }
13
14
15 void deleteCharInString(char str[], char c)
16 {
17     int i=0, j;
18
19     while (str[i])
20         if (str[i]==c)
21             for (j=i; str[j]; j++)
22                 str[j] = str[j+1];
23         else
24             i++;
25 }
26
27
28 int main()
29 {
30     char str[100];
31
32     getString(str);
33     deleteCharInString(str, getchar());
34     printf("%s", str);
35
36     return 0;
37 }
38

```

```
1  #include <stdio.h>
2
3
4  void string_filter(char s[])
5  {
6      int i, j, k;
7
8      for (i=0; s[i]; i++) {
9          for (j=1; (j<9)&&(s[i+j]==s[i]); j++);
10         if (j>1)
11             s[i++] = '0' + j;
12         if (j>2)
13             for (k=i+1; s[k]=s[k+j-2]; k++);
14     }
15 }
16
17
18 int main()
19 {
20     char s[31];
21
22     fgets(s, 31, stdin);
23     string_filter(s);
24     printf("%s", s);
25
26     return 0;
27 }
28
```