一:

1. D
2. C
3. A
4. C
5. D
6. B
7. A
8. B
9. C
10. A

二:

1.

b,B,A,b

2.

25

3.

285.00

4.

1000 10

5.

5

三:

1.

(1)

```
MAX
```

(2)

```
i=1
```

(3)

```
a[i]
```

(4)

```
i--
```

(5)

```
i!=0 //或者i>0
```

2.

(6)

```
#include<math.h>
```

(7)

```
return
```

(8)

```
simpson(0,2,n,f,&s)
```

(9)

```
simpson(0,PI/4,n,tg,&s)
```

(10)

```
fabs(x1-x)>eps
```

四:

1.

```
char cap(char c){
 int p='a'-'A';
 if (c>='a'&&c<='z')
 return c-p;
 else
 return c;
}
```

2.

```
#include<stdio.h>
float p(int n, float x){
 if (n==0)
 return 0;
 else if(n==1)
 return x;
 else
 return ((2*n-1)*x*p(n-1,x)-(n-1)*p(n-2,x))/n;
}
int main() {
 float rs = p(4,1.5);
 printf("%f\n",rs);
 return 0;
}
```

3.

```c
#include <stdio.h>
#include <math.h>
int main() {
 for (int i = 1; i < 1000; ++i) {
 int sum = 1;
 for (int j = 2; j < sqrt(i); ++j) {
 if (i%j==0)
 sum += j+i/j;
 }
 if (sum == i)
 printf("%d\n", i);
 }
 return 0;
}
```

4.

```c
#include <stdio.h>
int main() {
 float len =0,high=1000;
 for (int i = 0; i < 10; ++i) {
 len+=high;
 high=high/2;
 len+=high;
 }
 len-=high;
 printf("第十次落地时,共经过%fm\n", len);
 printf("第十次反弹%fm\n", high);
 return 0;
}
```

5.

```c
#include <stdio.h>
#include <math.h>
int main() {
 char string[1000]; //字符串数组
 int n=0; //字符数组长度
 char ch;
 ch = getchar();
 while (ch!='\n'){
 string[n] = ch;
n++;
ch = getchar();
}
string[n] = '\0';
int integers[1000]; //存储结果整数
int k=0,p=0,index=0,num=0,flag=0; //k,p分别为为字符串整数始末坐标，index为整数数组长度
for (int i = 0; i <= n; ++i) {
if (string[i]>='0'&&string[i]<='9'){
if (flag==0){ //该数字字符是第一次出现
flag=1;
k=i;
p=i;
} else{
p++;
}
} else{
if (flag==1){ //连续数字结束
num=0;
flag=0;
int c=0;
for (int j = p; j >=k ; j--) {
num+=(string[j]-'0')*(int)pow(10,c);
c++;
}
integers[index++] = num;
}
}
}
printf("共计%d个整数\n", index);
for (int l = 0; l < index; ++l) {
printf("%d\t", integers[l]);
}
return 0;
}
```