



학습목표

- ▶ 소수 구하는 알고리즘을 이해하고 프로그램을 작성할 수 있다.
- ▶ 최대 공약수, 진수 변환, 최댓값을 구하는 프로그램을 작성할 수 있다.



EBS tip

● 소수

약수가 1과 그 자신뿐이며, 1보다 큰 자연수이다.

1 소수를 구하는 프로그램

(1) 나눗셈을 이용한 소수 구하기

- ① 소수는 1과 그 자신으로 밖에 나누어지지 않는 2 이상의 자연수이다.
- ② 나눗셈을 이용하여 소수를 구하는 방법은 다음과 같다.
 - 2를 2로 나누어 본다. 2로만 나누어지므로 소수이다. 2를 출력한다.
 - 3을 2로 나누어 본다. 2로 나누어지지 않으므로 3으로 나누어 본다. 3으로만 나누어지므로 소수이다. 소수 3을 출력한다.
 - 4를 2로 나누어 본다. 2로 나누어지므로 소수가 아니다.
 - 5를 2, 3, 4로 나누어 본다. 2, 3, 4로 나누어지지 않고, 5로만 나누어지므로 소수이다. 소수 5를 출력한다.
 - 6을 2로 나누어 본다. 2로 나누어지므로 소수가 아니다.
 - 위와 같은 방법으로 어떤 수 N을 2부터 N-1까지의 정수로 나누어지면 소수가 아니고 나누어지지 않으면 소수이다.

예 소수 구하는 프로그램

프로그램 소스 코드	실행 결과
<pre> #include <stdio.h> void main() { int n, a, flag=0; printf("입력할 수="); scanf("%d", &n); for(a=2; a<n; a++) if (n%a==0){ flag=1; break; } if (flag==0) printf("%d => 소수\n", n); else printf("%d => 소수 아님\n", n); } </pre>	<p>입력할 수 = 13 13 => 소수</p>

[프로그램 설명]

- 소수 여부를 판단할 수 n을 입력받음.
- 2부터 n-1까지 변화하는 수를 저장할 변수 a 선언
- n을 a로 나누어 나머지가 0이면 나누어지는 수이므로 소수가 아니다. 따라서 이때는 소수를 판단하는 변수 flag를 1로 바꾸고 반복 루프를 빠져 나간다. a가 n보다 작으면 변수 a를 1씩 증가하면서 계속 소수 여부를 판단한다. 만일 a가 n과 같으면 n은 a로 나눈 나머지가 0이 아니므로 소수이다.



(2) 에라토스테네스의 체 방법을 이용한 소수 구하기

다음과 같이 1에서 30까지의 수 중에서 에라토스테네스의 체 방법을 통해 소수를 구하는 과정을 알아보자.

㉠ 1은 소수가 아니므로 지운다.

X	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

㉡ 2에 ○표시를 하고, 2를 제외한 2의 배수를 찾아 모두 지운다.

X	②	3	X	5	X	7	X	9	X
11	X	13	X	15	X	17	X	19	X
21	X	23	X	25	X	27	X	29	X

㉢ 다음 수가 지워지지 않은 수이면 그 수인 3에 ○표시를 하고, 3을 제외한 3의 배수를 찾아 모두 지운다.

X	2	③	X	5	X	7	X	X	X
11	X	13	X	X	X	17	X	19	X
X	X	23	X	X	X	X	X	29	X

㉣ 다음 수 4는 이미 지워졌으므로 그 다음 수 5를 기준으로 5을 제외한 5의 배수를 모두 지운다.

X	X	3	4	⑤	X	7	X	X	X
11	X	13	X	X	X	17	X	19	X
X	X	23	X	X	X	X	X	29	X

㉤ 이러한 방법으로 30까지 기준 수를 증가시키면서 배수가 없어질 때까지 반복한다.

X	2	3	X	5	X	⑦	X	X	X
11	X	13	X	X	X	17	X	19	X
X	X	23	X	X	X	X	X	29	X

㉦ 기준 수가 30에 도달했을 때 남은 수가 소수가 된다.

1	②	③	4	⑤	6	⑦	8	9	10
⑪	12	⑬	14	15	16	⑰	18	⑲	20
21	22	⑳	24	25	26	27	28	㉑	30

예 소수를 구하는 프로그램

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main() { int a, b, su[31]; for(a=0; a<=30; a++) su[a]=0; for(a=2; a<=30; a++) if (su[a]==0) { printf("%d ", a); for(b=2; b*a<=30; b++) su[a*b]=1; } }</pre>	2 3 5 7 11 13 17 19 23 29

[프로그램 해설]

㉠ 배열의 모든 요소 값에 0을 기억한다.

```
for(a=0; a<=30; a++)
    su[a]=0;
```

㉡ 첨자를 2부터 30까지 1씩 증가하면서, 그 첨자의 배열 요소의 값이 0이면, 첨자인 2를 출력하고, 첨자가 2의 배수인 배열 요소 값에 1을 기억한다.

```
for(a=2; a<=30; a++)
    if (su[a] == 0){
        printf("%d", a);
        for(b=2; b*a<=30; b++)
            su[a*b]=1;
    }
```

㉢ 위의 단계 ㉡를 반복한다.

2 최대 공약수를 구하는 프로그램

(1) 뺄셈을 이용하여 최대 공약수 구하기

최대 공약수를 구하고자 하는 두 수를 입력받아 큰 수에서 작은 수의 차가 0이면 그때의 수가 최대 공약수이다.

예

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main() { int x, y, t; printf("최대 공약수를 구할 두 수="); scanf("%d, %d", &x, &y); while(x){ if (x<y){ t = x; x = y; y = t; } x = x - y; } printf("최대 공약수 => %d\n", y); }</pre>	<p>최대 공약수를 구할 두 수 = 12, 48 최대 공약수=> 12</p>

(2) 나눗셈을 이용한 최대 공약수 구하기(유클리드 호제법)

뺄셈을 이용한 최대 공약수를 구하는 것은 두 수의 차가 클 때, 여러 번 반복 수행을 해야 한다. 그런데 이 뺄셈을 구하는 과정을 나머지 연산으로 바꾸면, 반복 횟수를 줄일 수 있다.

예

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main() { int x, y, t; printf("최대 공약수를 구할 두 수="); scanf("%d, %d", &x, &y); do{ t = x%y; x = y; y = t; } while(y); printf("최대 공약수 => %d\n", x); }</pre>	<p>최대 공약수를 구할 두 수 = 12, 48 최대 공약수 => 12</p>

[프로그램 해설]

- ㉠ 두 수를 입력받아 변수 x와 y에 저장한다.
㉡ x를 y로 나눈 나머지를 구하고, 두 수를 나머지와 제수로 바꾼다.

```
do{
    t=x%y;
    x=y;
    y=t;
}while(y);
```

- ㉢ 나머지가 0이 될 때까지 과정 ㉡를 반복하였을 때 변수 x의 값이 최대 공약수이다.



EBS tip

● 공약수

두 개(또는 그 이상)의 자연수에서 공통인 약수

● 최대 공약수

공약수 중에서 가장 큰 수

예 수학에서 최대 공약수 구하기

$$\begin{array}{r} 2) 12 \quad 48 \\ 2) \quad 6 \quad 24 \\ 3) \quad 3 \quad 12 \\ \quad 1 \quad 4 \end{array}$$

$$\text{최대 공약수} = 2 \times 2 \times 3 = 12$$

● 유클리드 호제법

두 자연수의 최대 공약수를 구하는 알고리즘의 하나로, 호제법이란 말은 두 수가 서로 상대방 수를 나누어서 결국 원하는 수를 얻는 알고리즘을 말한다.



3 진수로 변환하는 프로그램

(1) 10진수를 2진수로 변환하기

10진수를 2진수로 변환하려면, 10진수를 2로 나누어질 때까지 계속 나누어 각 단계에서 발생한 나머지를 나온 결과의 역순으로 표기한다.

$$\begin{array}{r}
 2 \overline{) 24} \\
 \underline{2 \overline{) 12}} \quad \cdots 0 \\
 \underline{2 \overline{) 6}} \quad \cdots 0 \\
 \underline{2 \overline{) 3}} \quad \cdots 0 \\
 \underline{1} \quad \cdots 1
 \end{array}
 \quad \begin{array}{l}
 11000 \\
 \text{그러므로 } (24)_{10} = (11000)_2
 \end{array}$$

▲ 10진수를 2진수로 바꾸는 과정

예

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main() { int num[100], n, a, b; printf("10진수 ="); scanf("%d", &n); a = 0; while(n > 0){ num[a] = n % 2; n = n / 2; a++; } printf("2진수 =>"); for(b=a-1; b>=0; b--){ printf("%d", num[b]); } }</pre>	<p>10진수 = 23 2진수 => 10111</p>

[프로그램 해설]

㉠ 변수 선언

- num[100]: 10진수를 2진수로 바꾼 수를 저장할 배열을 선언한다.
- a: 10진수를 2진수로 바꿀 때 몇 자리인가를 저장할 변수를 선언한다.

㉡ 2진수를 구하는 과정

```
while(n > 0){
    num[a] = n % 2;
    n = n / 2;
    a++; }
```

- num[a]=n%2 : 10진수 n을 2로 나눈 나머지를 배열에 저장한다.
- n=n/2 : 10진수 n을 2로 나눈 값을 다시 n 값으로 한다.
- a++ : a를 1씩 증가시킨다.

㉢ 구한 2진수를 출력하는 과정

```
for(b=a-1; b>=0; b--){
    printf("%d", num[b]);
}
```

(2) 10진수를 n진수로 변환하기(단, n은 16까지)

- ① 2진수 : 0과 1로 표현한다.
- ② 8진수 : 0, 1, 2, ..., 7의 8가지 종류의 숫자로 표현한다.

- 2진수 101은 다음과 같이 표현할 수 있다.

$$101 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

- 진수를 분명히 나타내기 위하여 다음과 같이 밑수를 첨자 형태로 사용하여 표현한다.

$$10110_{(2)}, 55_{(8)}, 2D_{(16)}$$

③ 16진수 : 0, 1, 2, ..., 9, A, B, C, D, E, F의 16가지 기호를 사용하여 표현한다.

예 10진수를 2진수부터 16진수까지 변환하는 프로그램

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main() { int num[100], n, k, a, b; char jin[17]="0123456789ABCDEF"; printf("변환할 진수="); scanf("%d", &k); printf("바꿀 10진수="); scanf("%d", &n); a = 0; while(n>0){ num[a] = n % k; n = n / k; a++; } printf("%d진수 => ", k); for(b=a-1; b>=0; b--) printf("%c", jin[num[b]]); }</pre>	<p>변환할 진수=16 바꿀 10진수=257</p> <p>16진수 => 101</p>

[프로그램 해설]

① 변수 선언

- num[100] : 10진수를 k진수로 바꾼 수를 저장할 배열을 선언한다.
- a는 10진수를 k진수로 바꿀 때 몇 자리인가를 저장할 변수로 0으로 초기화한다.

② k진수를 구하는 과정

```
while(n>0){
    num[a] = n % k;
    n = n / k;
    a++; }
```

- num[a]=n % k : 10진수 n을 변환할 k진수로 나눈 나머지를 배열에 저장한다.
- n=n/k : 10진수 n을 변환할 k진수로 나눈 값을 다시 n 값으로 한다.
- a++ : a를 1씩 증가시킨다.

4 최댓값을 구하는 프로그램

최댓값은 주어진 자료를 하나하나 비교하여 최댓값을 구하는 방법과 수와 관련된 프로그램을 작성한 후 구하는 방법이 있다. 여기서는 자료를 하나하나 비교하여 최댓값을 구하는 프로그램을 알아본다.

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void main() { int d[7]={12, 45, 23, 67, 34, 24, 33}; int a, m; m = d[0]; for(a=1; a<=6; a++) if (m<d[a]) m = d[a]; printf("최댓값=%d\n", m); }</pre>	<p>최댓값 = 67</p>



■ 구한 2진수를 출력하는 과정
for(b=a-1; b>=0; b--)
printf("%c", jin[num[b]]);

[프로그램 해설]

- 최댓값을 기억할 변수 m에 배열의 첫 번째 요소값을 저장한다.
- 배열의 첨자를 1부터 6까지 증가시키면서 나머지 요소값을 m 값과 비교하여 m 값보다 크면, m에 이 큰 값을 저장한다.



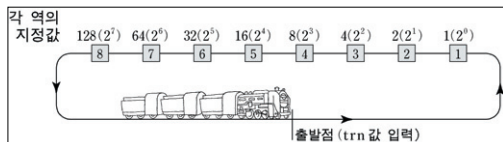
기출 모의고사

정답 및 해설 p. 18

1

2006학년도 대수능

다음은 8개 역(1~8)을 운행하는 화물 열차 제어 프로그램을 작성하기 위한 조건이다. 화물 열차 제어 프로그램 작성 과정에서 수의 진법 변환을 적용하였다. 점선 (가) 영역의 명령문으로 옳은 것은?



- 열차 제어 변수 trn의 입력값은 1~255의 정수이다.
- 제어 변수 trn 값에 따라 정지하는 역이 결정된다.
- 제어 변수 trn 값은 열차가 정지하는 역의 지정값들의 합과 같다.

예 trn=5이면 1번 역과 3번 역에 정지, trn=255이면 모든 역에 정지

```
#include <stdio.h>
void main() {
    int trn, idx, cnt;
    int flag[8]={0,0,0,0,0,0,0,0};
```

```
    printf("trn 입력[1~255]:");
    scanf("%d", &trn);
    idx = 0;
    while(trn>=2) {
        (가)
        trn = trn / 2;
        idx++;
    }
    flag[idx] = trn;
```

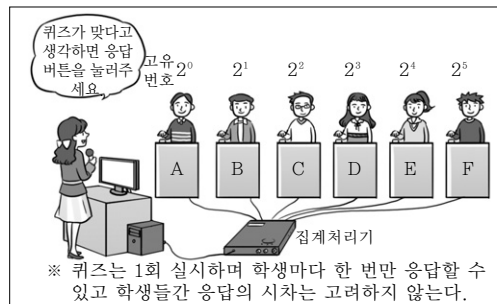
```
    printf("열차가 정지하는 역\n");
    for(cnt=0; cnt<=7; cnt++) {
        if (flag[cnt] == 1)
            printf("%d\n", cnt+1);
    }
}
```

- ① flag[idx] = trn - 2;
- ② flag[idx] = trn * 2;
- ③ flag[idx] = trn + 2;
- ④ flag[idx] = trn / 2;
- ⑤ flag[idx] = trn % 2;

2

2010학년도 대수능

그림은 응답버튼, 집계처리기, 프로그램으로 구성되어 [시스템의 동작 원리]에 따라 퀴즈 응답자를 판별하는 시스템이다. 이 시스템에 대한 설명으로 옳은 것만을 <보기>에서 있는 대로 고른 것은?



[시스템의 동작 원리]

- 응답버튼에는 고유번호가 지정되어 있다.
- 집계처리기는 눌러진 응답버튼의 고유번호를 합산하고, 이 값은 프로그램의 입력값(n)으로 사용된다.
- 프로그램은 입력되는 값(n)에 따라 응답버튼을 누른 학생 테이블에 해당하는 알파벳을 출력한다.

```
#include <stdio.h>
void main() {
    int a, b, n;
    char k[]={'A','B','C','D','E','F'};

    scanf("%d", &n);
    for(a=0; a<=2; a++) {
        for(b=0; b<=1; b++) {
            if(n%2==1) {
                printf("%c", k[a*2+b]);
            }
            (가) n=n/2;
        }
    }
}
```

<보기>

- ㄱ. 프로그램에 1이 입력되면 점선 (가)부분은 한 번 실행된다.
- ㄴ. 모든 학생이 응답버튼을 누르면 프로그램의 입력값(n)은 63이다.
- ㄷ. 테이블 A의 학생이 응답버튼을 누르면 프로그램의 입력값(n)은 홀수이다.

- ① ㄱ
- ② ㄴ
- ③ ㄱ, ㄷ
- ④ ㄴ, ㄷ
- ⑤ ㄱ, ㄴ, ㄷ

3 다음 프로그램을 실행한 결과 출력될 수 없는 값은?

```
#include <stdio.h>
void main() {
    int p[101], i, j;
    for(i=2; i<=100; i++)
        p[i]=i;
    for(i=2; i<=100; i++) {
        if (p[i]!=0)
        {
            printf( "%d \n", p[i] );
            for(j=i; j<=100; j+=i)
                p[j]=0;
        }
    }
}
```

- ① 5 ② 7 ③ 11 ④ 15 ⑤ 19

4 다음 프로그램의 실행 결과로 출력되는 변수 sum의 의미는?

```
#include <stdio.h>
void main() {
    int sw, sum, a, b;
    sum=2;
    for(b=3; b<=10; b++)
    {
        sw=0;
        for(a=2; a<=b-1; a++){
            if(b%a==0){
                sw=1;
                break;
            }
        }
        if (sw==0)
            sum=sum+b;
    }
    printf("%d", sum);
}
```

- ① 3부터 10까지의 소수
 ② 2부터 10까지 소수의 합
 ③ 2부터 10까지 홀수의 합
 ④ 3부터 10까지 짝수의 합
 ⑤ 1부터 10까지 자연수의 합

5 다음은 진수 변환에 관련된 알고리즘이다. 어떤 진수 변환 과정 알고리즘인가?

- (1) 변수 k=1, y=0으로 초기화한다.
- (2) 변환할 수 n을 입력받는다.
- (3) n을 10으로 나눈 나머지를 r에 대입한다.
- (4) y+r*k의 결과를 다시 y에 대입한다.
- (5) n을 10으로 나눈 몫을 다시 n에 대입한다.
- (6) k*8을 다시 k에 대입한다.
- (7) n이 0보다 크면 (3)으로 이동한다.
- (8) y를 출력한다.

- ① 8진수를 2진수로 ② 8진수를 10진수로
 ③ 8진수를 16진수로 ④ 10진수를 8진수로
 ⑤ 16진수를 8진수로

6 다음은 입력한 두 정수의 최대 공약수를 구하는 프로그램이다. (가) 안에 들어갈 내용으로 옳은 것은?

```
#include <stdio.h>
void main() {
    int su1, su2, i, gcd;
    scanf("%d,%d", &su1, &su2);
    gcd=1;

    for(i=1; i<=su1; ++i){
        if ( (가) )
            gcd=i;
    }
    printf("최대 공약수=%d\n", gcd);
}
```

- ① su1 % i == 0 || su2 % i == 0
 ② su1 % i == 0 && su2 % i == 0
 ③ su1 % i == 1 && su2 % i == 1
 ④ su1 % i != 0 || su2 % i != 0
 ⑤ su1 % i >= 0 && su2 % i >= 0