

TEAM EX1-1 배열로 4가지 수의 합 구하기

```
01 #include <iostream>
02 using namespace std;
03 int main( )
04 {
05     int aa[4];
06     int sum=0;
07     int i;
08
09     for (i=0 ; i<=3 ; i++)
10     {
11         cout<<"Enter the "<< i+1<<"th number : ";
12         cin>> __①__ ;
13     }
14
15     sum = aa[0] + aa[1] + aa[2] + aa[3];
16
17     cout<<" Sum : "<<sum<<endl;
18 }
```

TEAM EX1-2 배열로 4가지 수의 합 구하기

```
01 #include <iostream>
02 using namespace std;
03 int main( )
04 {
05     int aa[4];
06     int sum=0;
07     int i;
08
09     for (i=0 ; i<=3 ; i++)
10     {
11         cout<<"Enter the "<< i+1<<"th number : ";
12         cin>> __①__ ;
13     }
14
15     sum = aa[0] + aa[1] + aa[2] + aa[3]; //for 문을 사용하여 4번의 덧셈을 수행하도록 수정해보기
16
17     cout<<" Sum : "<<sum<<endl;
18 }
```

TEAM EX2 배열 접근 연습

- for 문을 사용한 배열 초기화
 - ① aa[100]에 2의 배수로 초기화하기
 - ② bb[100]에 역순으로 넣기

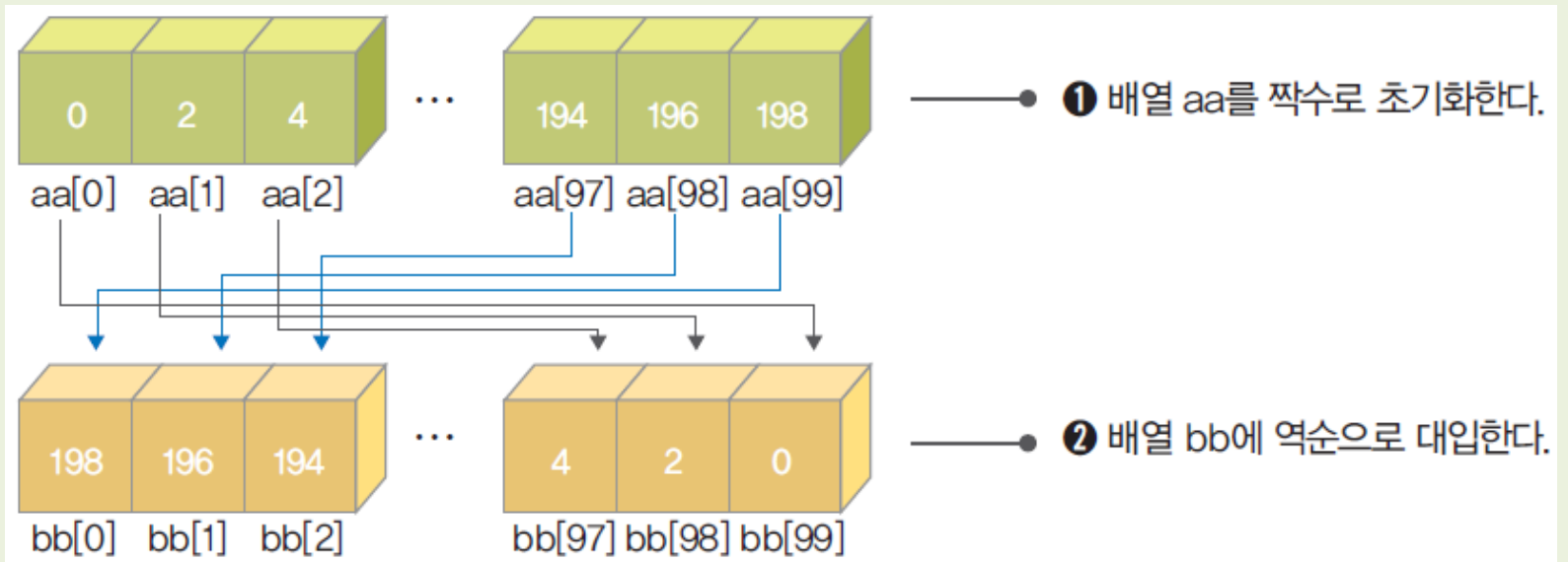


그림 8-7 배열의 초기화 ⑤

TEAM EX2 배열 접근 연습

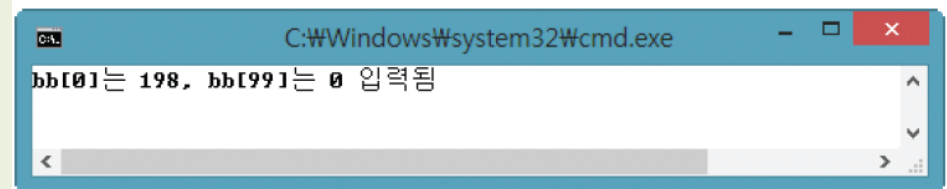
```
01 #include <iostream>
02 using namespace std;
03 int main( )
04 {
05     int aa[100], bb[100];
06     int i;
07
08     for (i=0 ; i<100 ; i++)
09     {
10         aa[i] = i * 2;
11     }
12
13     for (i=0 ; i<100 ; i++)
14     {
15         bb[i] = ____①____ ;
16     }
17
18     cout<<"bb[0]는"<< bb[0]<<" , bb[99]는 "<<bb[99] <<"입력됨"<<endl;
19 }
```

---배열 aa와 bb를 선언한다.

---배열 aa[0], aa[1], aa[2], ..., aa[99]에
0, 2, 4, ..., 198이 대입된다

---배열 bb[0]에 aa[99]를 대입하는 방식으로
값이 역순으로 대입된다.

실행결과 ▼

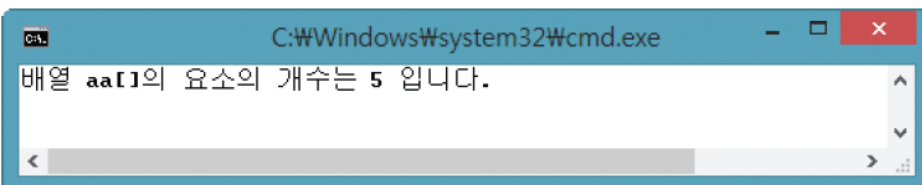


TEAM EX3 배열 크기 계산해보기

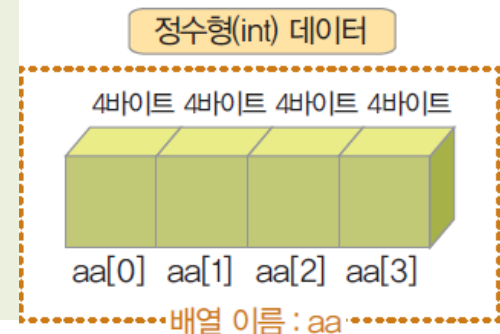
```
01 #include <iostream>
02 using namespace std;
03 int main( )
04 {
05     int aa[] = {10, 20, 30, 40, 50};    ---배열을 선언한다. 변수의 개수를 지정하지 않고 초기화
06     int count;                          ---배열 크기를 저장할 변수
07
08     count = sizeof(aa) / sizeof(int);    ---배열 크기를 계산한다.
09
10     cout<<"배열 aa[]의 요소의 개수는 "<<count<<" 입니다.\n";
11 }
```

배열의 크기(요소 개수) = sizeof(전체 배열 이름) / sizeof(배열의 데이터 형식);

실행결과 ▼



```
C:\Windows\system32\cmd.exe
배열 aa[]의 요소의 개수는 5 입니다.
```



↑ 16바이트 = 4바이트 × 4

TEAM EX4 구구단의 결과를 2차원 배열에 저장

```
01 #include <iostream>
02 using namespace std;
03 int main( )
04 {
05     int gugu[9][9];
06     int i, k;
07
08     for(i=0 ; i<9 ; i++)
09         for (k=0 ; k<9 ; k++)
10             gugu[i][k] = (i+1) * (k+1) ;
11
12     for(i=0 ; i<9 ; i++)
13     {
14         for (k=0 ; k<9 ; k++)
15         {
16             cout<<k+1<<"X"<<i+1<<"="<<gugu[i][k];
17         }
18     }
19 }
```

----문자형 2차원 배열 gugu와 첨자 변수 i, k를 선언한다.

----구구단을 곱한 결과를 2차원 배열에 저장한다.
i, k가 0부터 시작되므로 1을 더해서 곱했다.

----구구단 결과를 출력한다.

```
C:\Windows\system32\cmd.exe
1X1= 1 2X1= 2 3X1= 3 4X1= 4 5X1= 5 6X1= 6 7X1= 7 8X1= 8 9X1= 9
1X2= 2 2X2= 4 3X2= 6 4X2= 8 5X2=10 6X2=12 7X2=14 8X2=16 9X2=18
1X3= 3 2X3= 6 3X3= 9 4X3=12 5X3=15 6X3=18 7X3=21 8X3=24 9X3=27
1X4= 4 2X4= 8 3X4=12 4X4=16 5X4=20 6X4=24 7X4=28 8X4=32 9X4=36
1X5= 5 2X5=10 3X5=15 4X5=20 5X5=25 6X5=30 7X5=35 8X5=40 9X5=45
1X6= 6 2X6=12 3X6=18 4X6=24 5X6=30 6X6=36 7X6=42 8X6=48 9X6=54
1X7= 7 2X7=14 3X7=21 4X7=28 5X7=35 6X7=42 7X7=49 8X7=56 9X7=63
1X8= 8 2X8=16 3X8=24 4X8=32 5X8=40 6X8=48 7X8=56 8X8=64 9X8=72
1X9= 9 2X9=18 3X9=27 4X9=36 5X9=45 6X9=54 7X9=63 8X9=72 9X9=81
```

오른쪽과 같은 실행 결과가 나오도록 하려면??