

07. Quiz

Implementations ^[1]

- **Structure Set**
 - 정수형의 변수 'size'를 가진다.
 - 정수형 포인터 변수 'values'을 가진다.
- **Structure Pair**
 - 정수형의 변수 'x' 'y'를 가진다.
- **Structure Cartesian**
 - 정수형의 변수 'size'를 가진다.
 - Pair 타입의 포인터 배열 'values'을 가진다.
- 0/1을 TRUE/FALSE 로 정의하여 사용한다.

Implementations [2-1]

- **Declarations**

```
typedef struct set
{
    int size;
    int *elems;
}Set;

typedef struct pair
{
    int x;
    int y;
}Pair;

typedef struct cartesian
{
    int size;
    Pair **values;
}Cartesian;
```

Implementations [2-2]

- Prototypes

```
int isExist(Set* set, int elem);
int getCardinality(Set* set);

void getUnion(Set* a, Set* b, Set** unionSet);
void getIntersection(Set* a, Set* b, Set** intersectionSet);
void getComplement(Set* u, Set* a, Set** complementSet);
void getDifference(Set* a, Set* b, Set** differenceSet);
void getSymmetricDifference(Set* a, Set** symDifferenceSet);
void getCartesianProduct(Set* a, Set* b, Cartesian** cartesianSet);

void printSet(Set* set);
void printCartesian(Cartesian* set);
void freeSet(Set* set);
void freeCartesian(Cartesian* set);
```

Implementations [2-3]

- Main

```
int main(int argc, char* argv[])
{
    int u[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    int a[] = {1, 2, 3, 4, 5};
    int b[] = {4, 5, 6, 7, 8};
    int c[] = {100, 200};

    Set setU = { (sizeof(u) / sizeof(int)), u };
    Set setA = { (sizeof(a) / sizeof(int)), a };
    Set setB = { (sizeof(b) / sizeof(int)), b };
    Set setC = { (sizeof(c) / sizeof(int)), c };

    Set* unionSet      = NULL;
    Set* intersectionSet = NULL;
    Set* complementSet1 = NULL;
    Set* complementSet2 = NULL;
    Set* differenceSet1 = NULL;
    Set* differenceSet2 = NULL;
    Set* symDifferenceSet = NULL;
    Cartesian* cartesianSet = NULL;
```

Implementations [2-4]

- Main

```
getUnion(&setA, &setB, &unionSet);
getIntersection(&setA, &setB, &intersectionSet);
getComplement(&setU, &setA, &complementSet1);
getComplement(&setU, &setB, &complementSet2);
getDifference(&setA, &setB, &differenceSet1);
getDifference(&setB, &setA, &differenceSet2);
getSymmetricDifference(&setA, &setB, &symDifferenceSet);
getCartesianProduct(&setA, &setC, &cartesianSet);
```

```
printf("          << Set Operations >> □n");
printf(" 1. Union A and B □n");
printf(" = " );
printSet(unionSet);
printf(" 2. Intersection A and B □n");
printf(" = " );
printSet(intersectionSet);
printf(" 3. Complement A □n");
printf(" = " );
printSet(complementSet1);
printf(" 4. Complement B □n");
printf(" = " );
printSet(complementSet2);
```

Implementations [2-5]

- Main

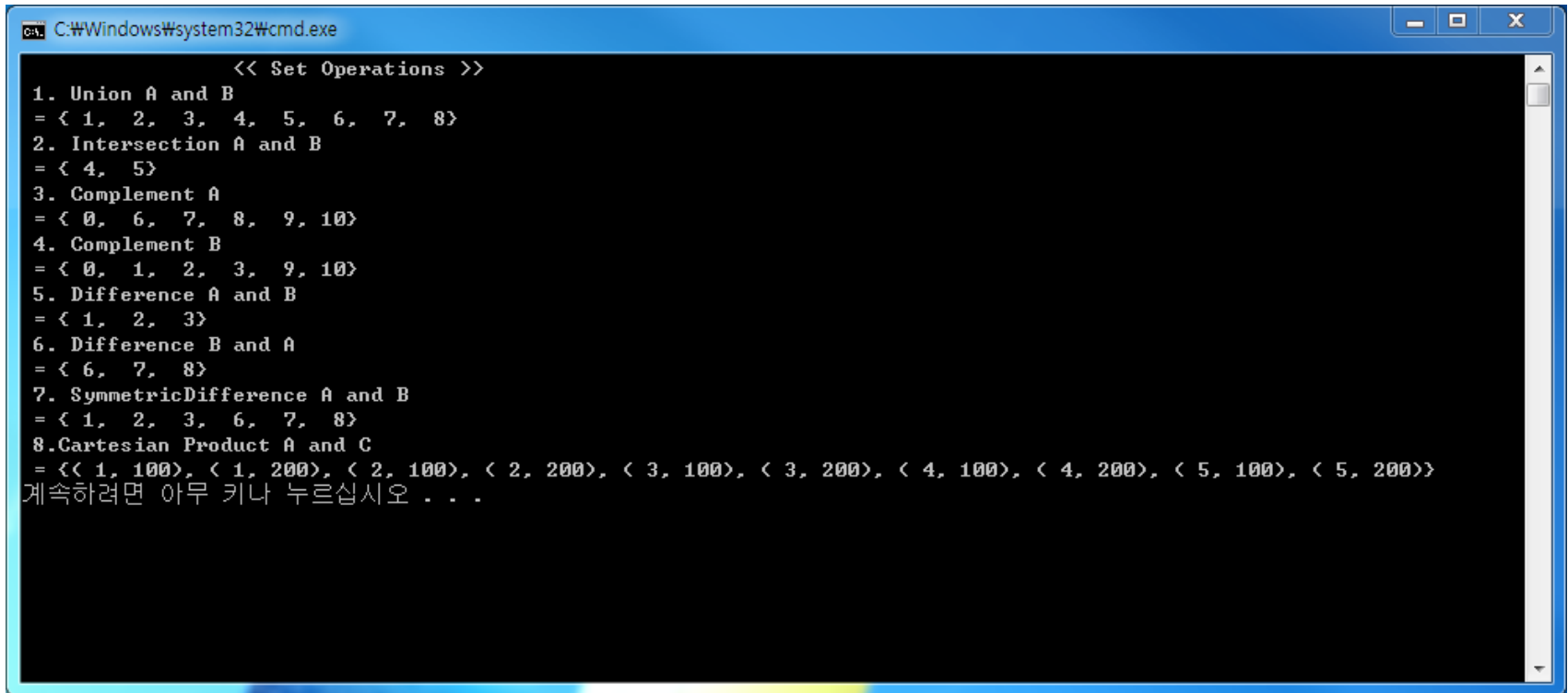
```
printf(" 5. Difference A and B\n");
printf(" = " );
printSet(differenceSet1);
printf(" 6. Difference B and A\n");
printf(" = " );
printSet(differenceSet2);
printf(" 7. SymmetricDifference A and B\n");
printf(" = " );
printSet(symDifferenceSet);
printf(" 8. Cartesian Product A and C\n");
printf(" = " );
printCartesian(cartesianSet);

freeSet(unionSet);
freeSet(intersectionSet);
freeSet(complementSet1);
freeSet(complementSet2);
freeSet(differenceSet1);
freeSet(differenceSet2);
freeSet(symDifferenceSet);
freeCartesian(cartesianSet);

return 0;
}
```

Implementations [3]

- Run



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window has a blue title bar and standard Windows window controls (minimize, maximize, close) in the top right corner. The command prompt shows a series of set operations and their results. The text is as follows:

```
<< Set Operations >>
1. Union A and B
= { 1, 2, 3, 4, 5, 6, 7, 8}
2. Intersection A and B
= { 4, 5}
3. Complement A
= { 0, 6, 7, 8, 9, 10}
4. Complement B
= { 0, 1, 2, 3, 9, 10}
5. Difference A and B
= { 1, 2, 3}
6. Difference B and A
= { 6, 7, 8}
7. SymmetricDifference A and B
= { 1, 2, 3, 6, 7, 8}
8. Cartesian Product A and C
= {< 1, 100>, < 1, 200>, < 2, 100>, < 2, 200>, < 3, 100>, < 3, 200>, < 4, 100>, < 4, 200>, < 5, 100>, < 5, 200>}
계속하려면 아무 키나 누르십시오 . . .
```