

# 2013 Fall Semester SMP

Day 2 – Data type with I/O, structure

# Contents

---

Data type with I/O

Function

Conditional statement

for statement

Pointer

Algorithm

---



# Contents

---

Data type with I/O	int
Function	char
Conditional statement	float
for statement	double
Pointer	structure
Algorithm	array



# Printf

---

- ▶ 단순히 출력을 하는 것은 문제없음.
  - ▶ 근데 조금 이쁘게 출력하고 싶으면??
  - ▶ 이쁘게 머지...



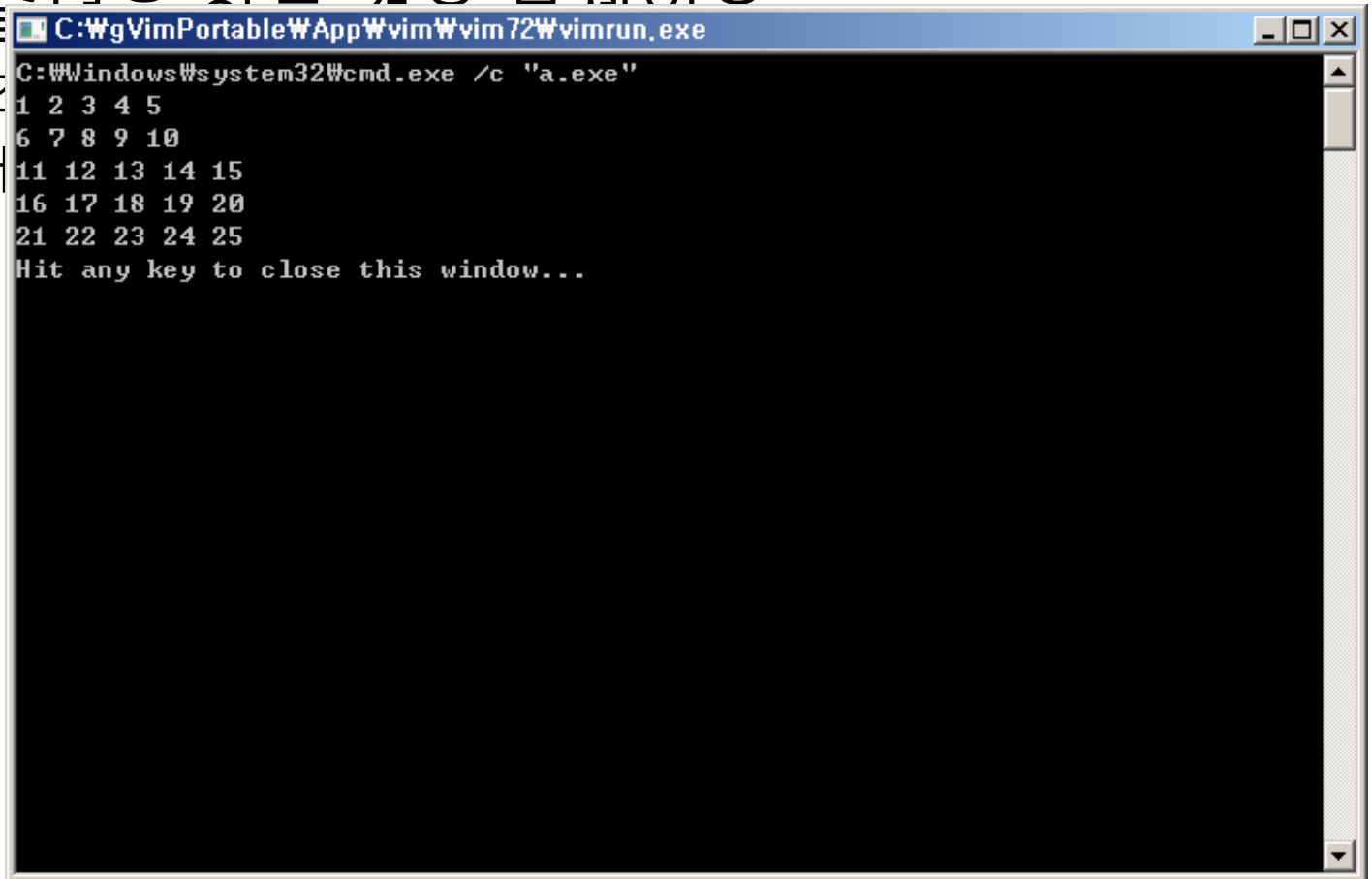
# Printf

---

▶ 단순히 출력하는 것만으로도

▶ 근데

▶ 이쁘게



```
C:\WgVimPortable\App\vim\vim72\vimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
Hit any key to close this window...
```



# Printf

---

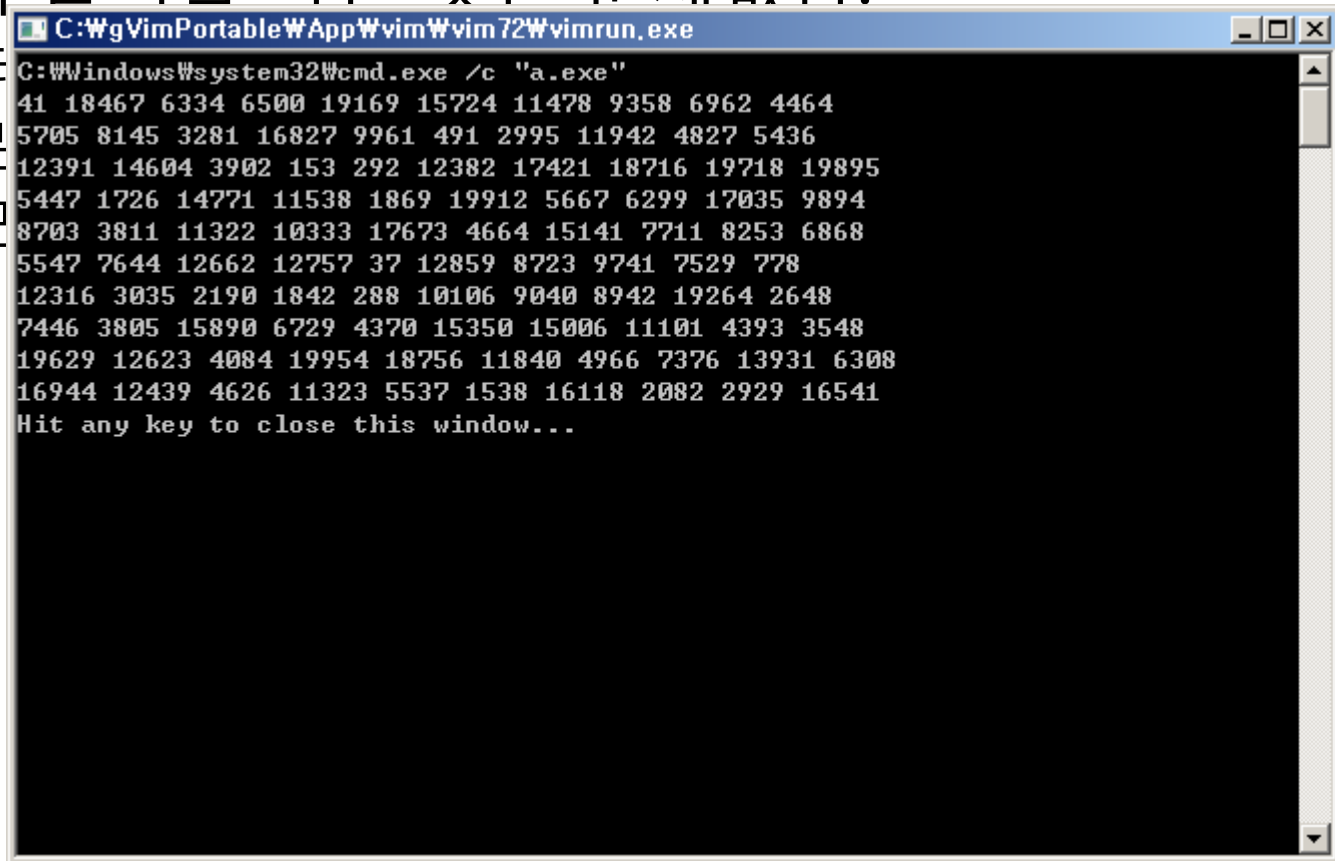
- ▶ 단순히 출력을 하는 것은 문제없음.
  - ▶ 근데 조금 이쁘게 출력하고 싶으면??
  - ▶ 이쁘게 머지...
  - ▶ 볼만한데?



# Printf

- ▶ 단순히 출력을 하는 것은 문제없음.

- ▶ 근대
- ▶ 이분
- ▶ 불분



```
C:\WgVimPortable\WApp\vim\vim72\vimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
41 18467 6334 6500 19169 15724 11478 9358 6962 4464
5705 8145 3281 16827 9961 491 2995 11942 4827 5436
12391 14604 3902 153 292 12382 17421 18716 19718 19895
5447 1726 14771 11538 1869 19912 5667 6299 17035 9894
8703 3811 11322 10333 17673 4664 15141 7711 8253 6868
5547 7644 12662 12757 37 12859 8723 9741 7529 778
12316 3035 2190 1842 288 10106 9040 8942 19264 2648
7446 3805 15890 6729 4370 15350 15006 11101 4393 3548
19629 12623 4084 19954 18756 11840 4966 7376 13931 6308
16944 12439 4626 11323 5537 1538 16118 2082 2929 16541
Hit any key to close this window...
```

# Printf

---

- ▶ 단순히 출력을 하는 것은 문제없음.
  - ▶ 근데 조금 이쁘게 출력하고 싶으면??
  - ▶ 이쁘게 머지...
  - ▶ ~~볼만한데?~~





# Printf

---

- ▶ 단순히 출력을 하는 것은 문제없음.
  - ▶ 근데 조금 이쁘게 출력하고 싶으면??
  - ▶ 이쁘게 머지...
  - ▶ ~~불만한테?~~
  - ▶ 어떠한 숫자의 좌표 (r, c) 를 바로 잡기 힘들



# Printf

## ▶ 단순히 추려오 하느 게으 르제어으

- ▶ 근
- ▶ 이
- ▶ 볼
- ▶ 어

```
C:\WgVimPortable\WApp\vim\vim72\vimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
  41  18467  6334  6500  19169  15724  11478  9358  6962  4464
 5705  8145  3281  16827  9961  491  2995  11942  4827  5436
12391 14604  3902  153  292  12382  17421  18716  19718  19895
 5447  1726  14771  11538  1869  19912  5667  6299  17035  9894
 8703  3811  11322  10333  17673  4664  15141  7711  8253  6868
 5547  7644  12662  12757  37  12859  8723  9741  7529  778
12316  3035  2190  1842  288  10106  9040  8942  19264  2648
 7446  3805  15890  6729  4370  15350  15006  11101  4393  3548
19629 12623  4084  19954  18756  11840  4966  7376  13931  6308
16944 12439  4626  11323  5537  1538  16118  2082  2929  16541
Hit any key to close this window...
```

# Printf

## ▶ 단순히 추려오 하느 게으 르제어으

- ▶ 근
- ▶ 이
- ▶ 볼
- ▶ 어

The image shows a Windows command prompt window with a blue title bar. The title bar text is "C:\WgVimPortable\WApp\vim\vim72\vimrun.exe". The command prompt shows the command "C:\Windows\system32\cmd.exe /c "a.exe"". Below the command, there is a table of numbers with 10 columns and 10 rows. The numbers are: 41, 18467, 6334, 6500, 19169, 15724, 11478, 9358, 6962, 4464; 5705, 8145, 3281, 16827, 9961, 491, 2995, 11942, 4827, 5436; 12391, 14604, 3902, 153, 292, 12382, 17421, 18716, 19718, 19895; 5447, 1726, 14771, 11538, 1869, 19912, 5667, 6299, 17035, 9894; 8703, 3811, 11322, 10333, 17673, 4664, 15141, 7711, 8253, 6868; 5547, 7644, 12662, 1275; 12316, 3035, 2190, 184; 7446, 3805, 15890, 672; 19629, 12623, 4084, 1995; 16944, 12439, 4626, 1132. Below the table, it says "Hit any key to close this w".

Overlaid on the bottom right of the command prompt is a video frame showing a group of young men. One man in the center is wearing a blue jacket and has a pained or shouting expression. A white box with the Korean text "중군?" is overlaid on the bottom right of the video frame.

# Printf

---

- ▶ 단순히 출력을 하는 것은 문제없음.
  - ▶ 근데 조금 이쁘게 출력하고 싶으면??
  - ▶ 이쁘게 머지...
  - ▶ ~~불만한테?~~
  - ▶ 어떠한 숫자의 좌표 (r, c) 를 바로 잡기 힘들



# Printf

---

```
int x = 372;
```

`%d`



# Printf

---

```
int x = 372;
```

`%d`

`> 372`

---



# Printf

---

```
int x = 372;
```

`%7d`



# Printf

---

```
int x = 372;
```

`%7d`

`> 372`





# Printf

---

```
int x = 372;
```

```
%7d
```



# Printf

---

```
int x = 372;
```

`%7d`



# Printf

---

```
int x = 372;
```

`%7d`

				3	7	2
--	--	--	--	---	---	---



# Printf

---

```
int x = 372, y = 273;
```

```
%7d%6d
```

>

---

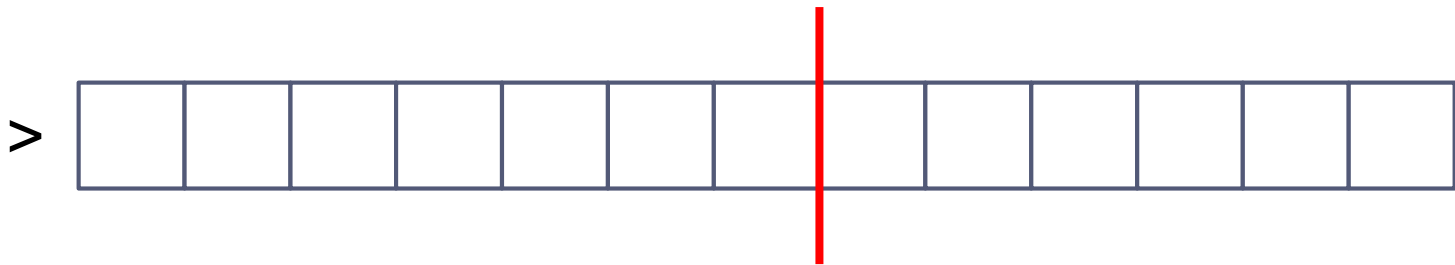


# Printf

---

```
int x = 372, y = 273;
```

`%7d%6d`

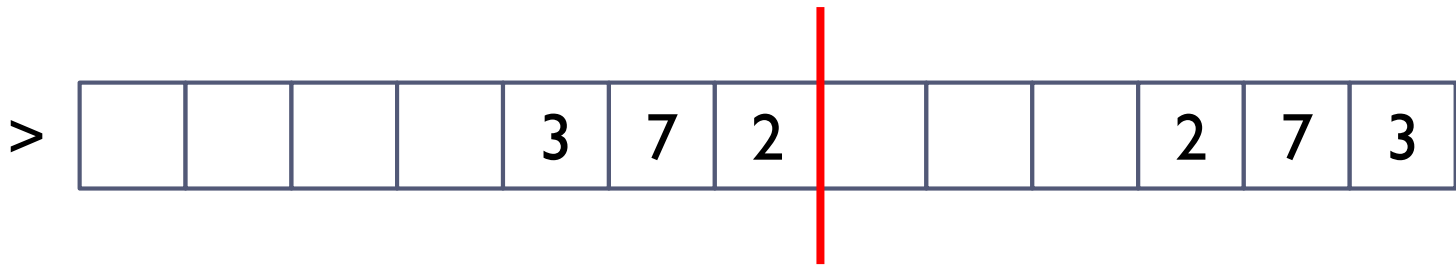


# Printf

---

```
int x = 372, y = 273;
```

`%7d%6d`



# Printf

---

```
int x = 372, y = 273;
```

`%7d%6d`

>

3 7 2

2 7 3

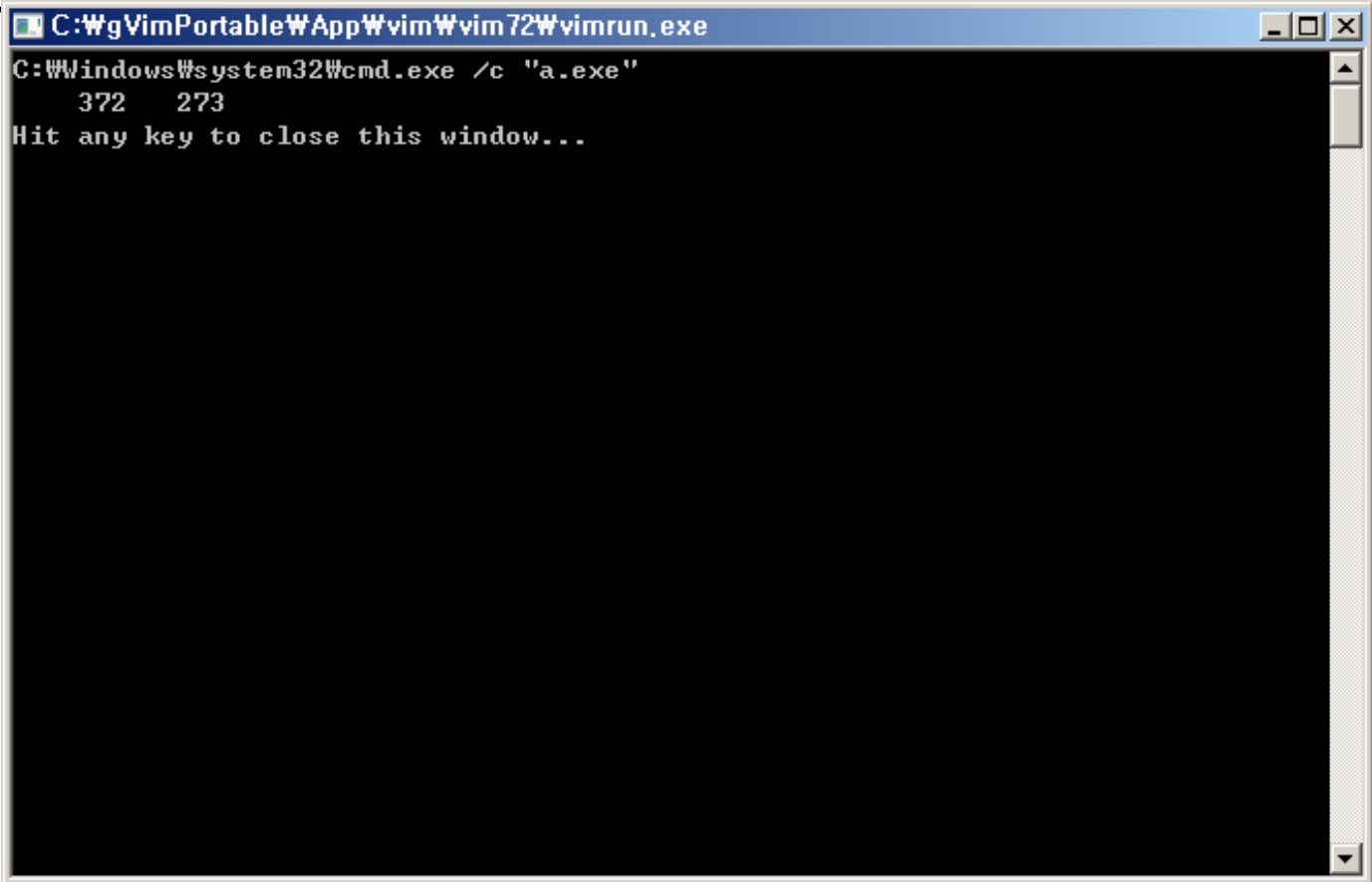
---



# Printf

---

```
int x = 372 - 273.
```



A screenshot of a Windows command prompt window. The title bar reads "C:\WgVimPortable\App\vim\vim72\vimrun.exe". The command prompt shows the execution of a program: "C:\Windows\system32\cmd.exe /c "a.exe"". The output of the program is "372 273" on the first line and "Hit any key to close this window..." on the second line. The window has a standard Windows interface with a title bar, maximize, minimize, and close buttons, and a scroll bar on the right.

v





# Printf

---

```
int x = 4473;
```

`%3d`

>



# Printf

---

```
int x = 4473;
```

`%3d`

>



# Printf

---

```
int x = 4473;
```

`%3d`



# Printf

---

```
int x = 4473;
```

`%3d`

> 

4	4	7
---	---	---

 3



# Printf

---

```
int x = 4473;
```

`%3d`

> 4 4 7 3



# Printf

---

```
int x = 4473;
```

```
%-7d
```

```
>
```



# Printf

---

```
int x = 4473;
```

`%-7d`

>



# Printf

---

```
int x = 4473;
```

`%-7d`

> 

4	4	7	3			
---	---	---	---	--	--	--





# Printf

---

```
int x = 4473;
```

`%-7d`

```
> 4 4 7 3
```

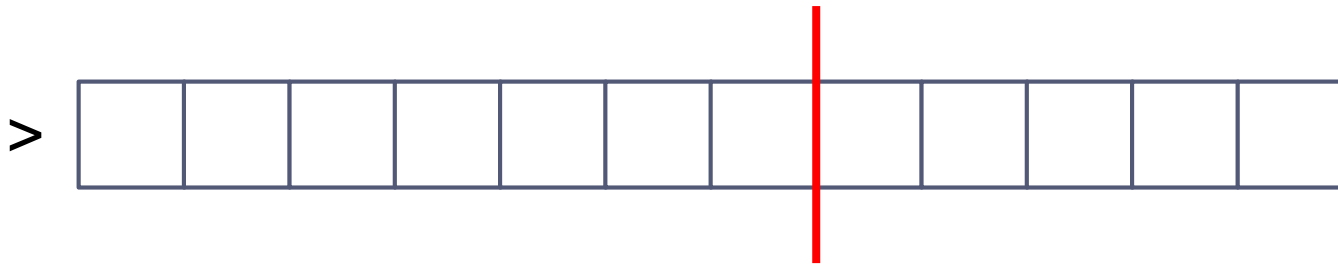


# Printf

---

```
int x = 4473, y = 112;
```

`%-7d%-5d`

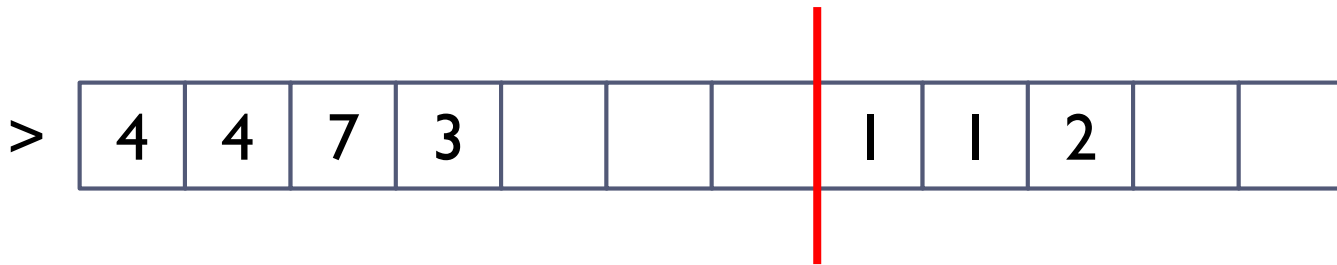


# Printf

---

```
int x = 4473, y = 112;
```

`%-7d%-5d`



# Printf

---

```
int x = 4473, y = 112;
```

```
%-7d%-5d
```

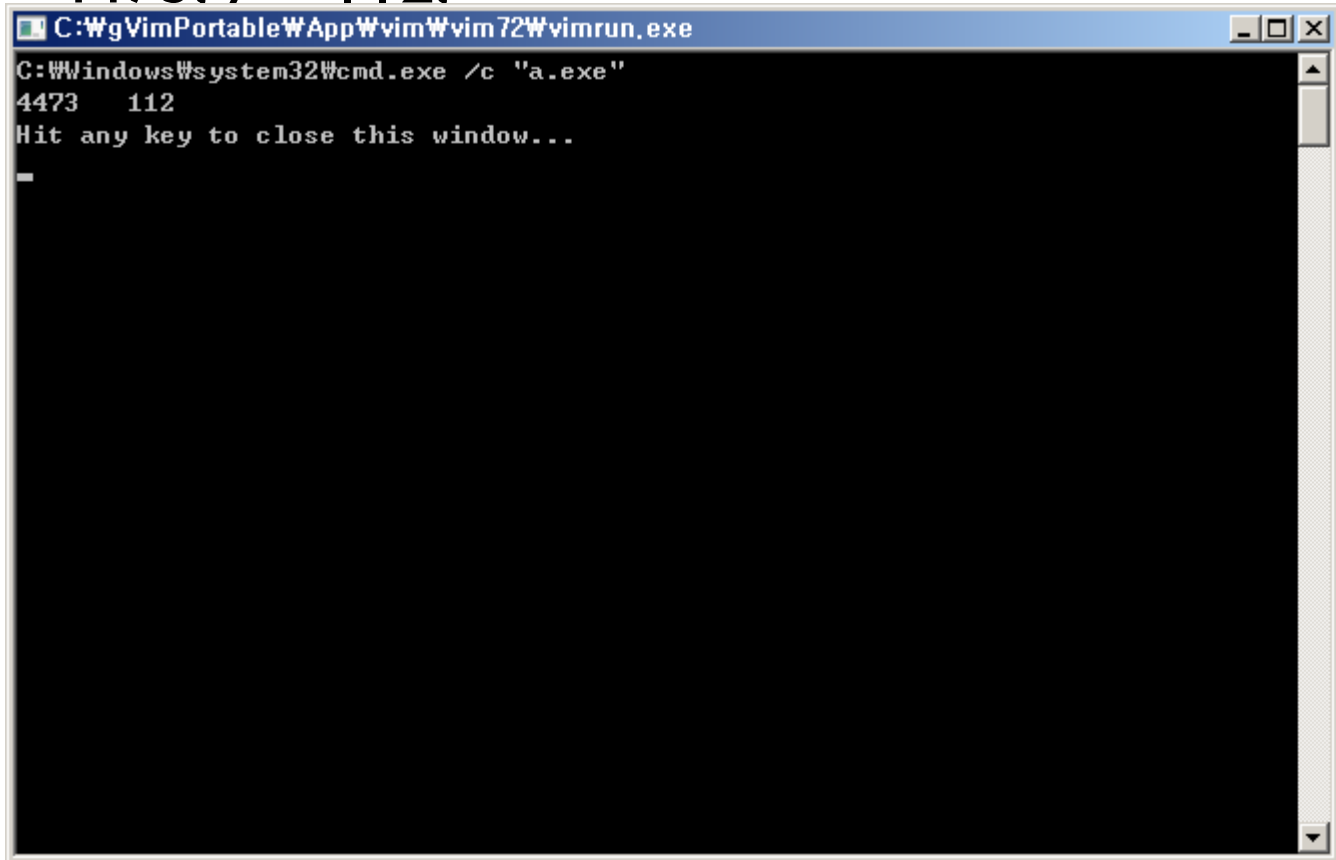
```
> 4 4 7 3           1 1 2
```



# Printf

---

int x = 4473, y = 112:



```
C:\WgVimPortable\App\vim\vim72\vimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
4473  112
Hit any key to close this window...
-
```

v



# Scanf

---

- ▶ 데이터를 입력 받을 때 사용
  - ▶ & 붙이지 않는 흔한 실수를 많이 함
  - ▶ Programming 초보자의 경우 프로그램이 갑자기 죽으면 대부분 & 문제
  - ▶ 이유는 아직 몰라도 됨
  - ▶ printf처럼 %와 d 사이에 숫자를 붙일 수 있긴 한데 일단 skip

```
scanf ( “%d %d”, &a, &b);
```

---

# Scanf

---

- ▶ 데이터를 입력 받을 때 사용
  - ▶ & 붙이지 않는 흔한 실수를 많이 함
  - ▶ Programming 초보자의 경우 프로그램이 갑자기 죽으면 대부분 & 문제
  - ▶ 이유는 아직 몰라도 됨
  - ▶ printf처럼 %와 d 사이에 숫자를 붙일 수 있긴 한데 일단 skip

```
scanf ( “%d %d”, &a, &b);
```



# Scanf

---

- ▶ 데이터를 입력 받을 때 사용
  - ▶ & 붙이지 않는 흔한 실수를 많이 함
  - ▶ Programming 초보자의 경우 프로그램이 갑자기 죽으면 대부분 & 문제
  - ▶ 이유는 아직 몰라도 됨
  - ▶ printf처럼 %와 d 사이에 숫자를 붙일 수 있긴 한데 일단 skip

```
scanf ( “%d %d”, &a, &b);
```





# Scanf

---

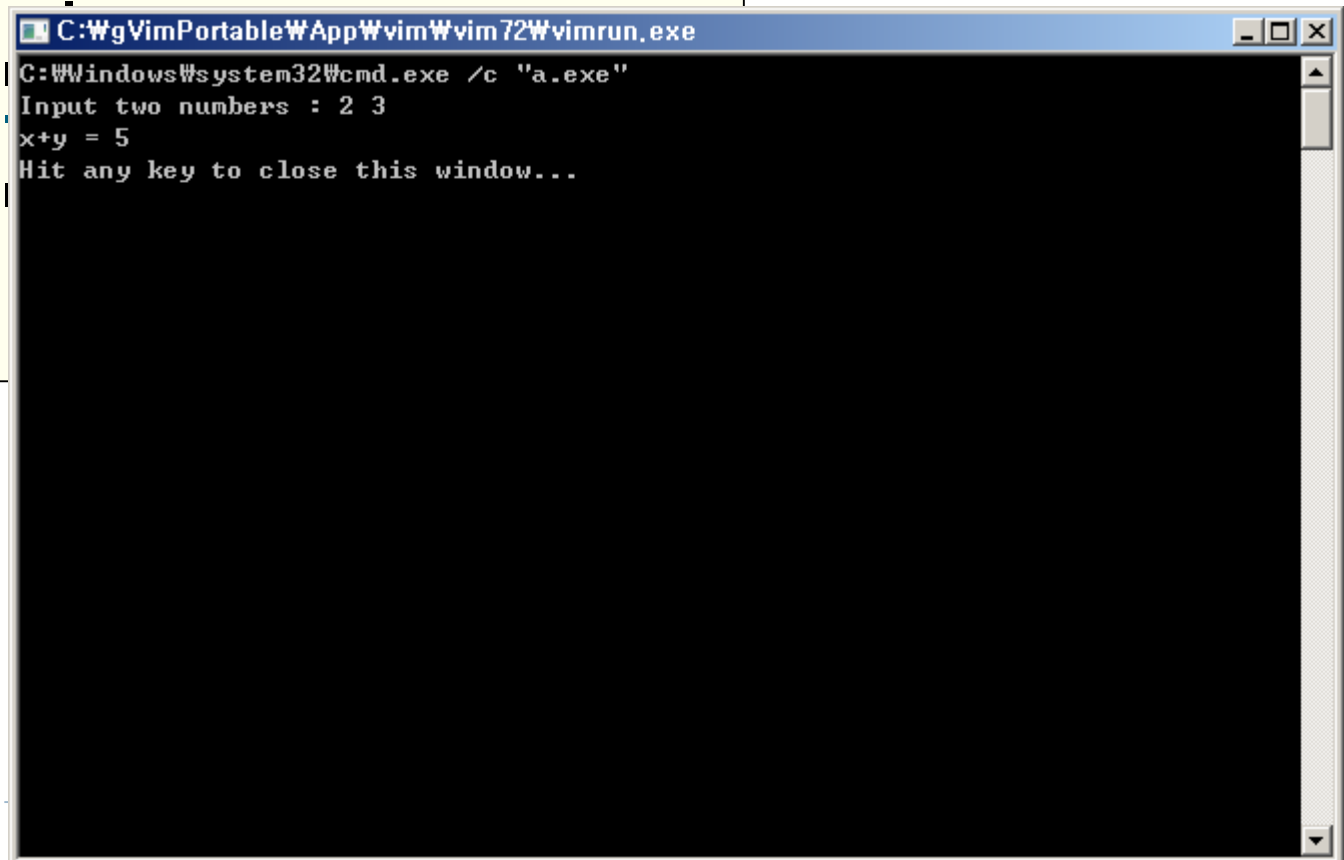
```
#include <stdio.h>
int main(){
    int x, y;
    printf("Input two numbers : ");
    scanf("%d %d",&x, &y);
    printf("x+y = %d\n",x+y);
    return 0;
}
```



# Scanf

---

```
#include <stdio.h>
int main(){
    int x, y;
    printf("Input two numbers : ");
    scanf("%d %d", &x, &y);
    printf("x+y = %d\n", x+y);
    return 0;
}
```



```
C:\WgVimPortableWAppWvimWvim72Wvimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
Input two numbers : 2 3
x+y = 5
Hit any key to close this window...
```

# Example

---

- ▶ 숫자를 하나를 입력받아 그대로 출력하는 프로그램
- ▶ 숫자 두개를 입력받아  $x-y$  를 출력하는 프로그램
- ▶ 2x2 행렬의 element를 입력받아 행렬 “이쁘게” 출력



# char

---

- ▶ char는 문자 하나를 담는 변수
  - ▶ 문자도 결국 하나의 숫자
  - ▶ 0 ~ 255 까지의 숫자를 문자로 대응을 시켜놓음
  - ▶ ASCII Table
  - ▶ Table의 숫자는 절대로 외울 필요 없음
  - ▶ 1 byte

A → 65



# Char Decl / Output

---

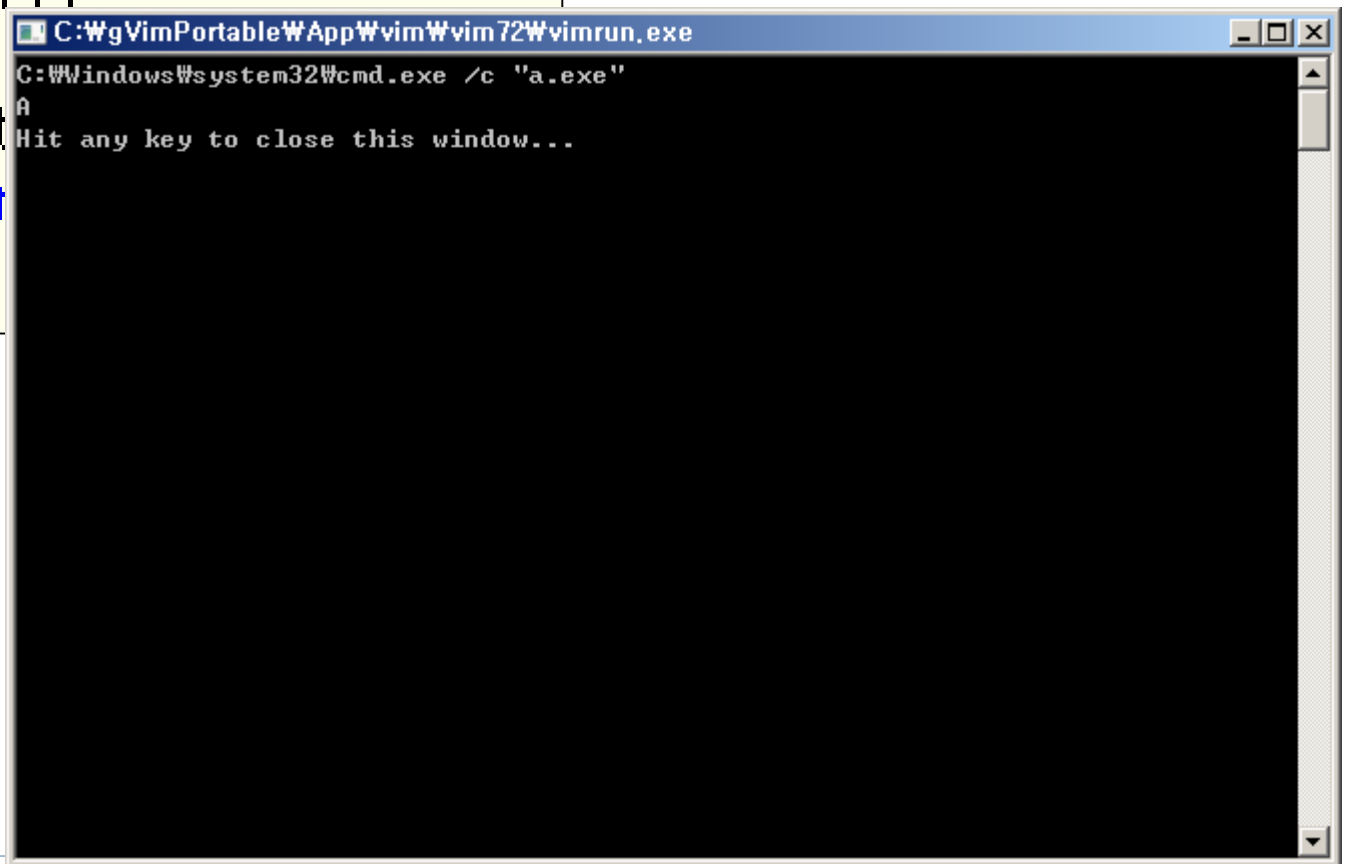
```
#include <stdio.h>
int main(){
    char x = 'A';
    printf("%c#\n", x);
    return 0;
}
```



# Char Decl / Output

---

```
#include <stdio.h>
int main()
{
    char
    print
    return
}
```



The screenshot shows a Windows command prompt window titled "C:\WgVimPortable\WApp\vim\vim72\vimrun.exe". The prompt is "C:\Windows\system32\cmd.exe /c "a.exe"". The output of the program is "Hit any key to close this window...".

# Char Decl / Output

---

`%c`



# Char Decl / Input

---

```
#include <stdio.h>
int main(){
    char x;
    printf("Input a character : ");
    scanf("%c",&x);
    printf("Your character : %c#\n",x);
    //
    return 0;
}
```

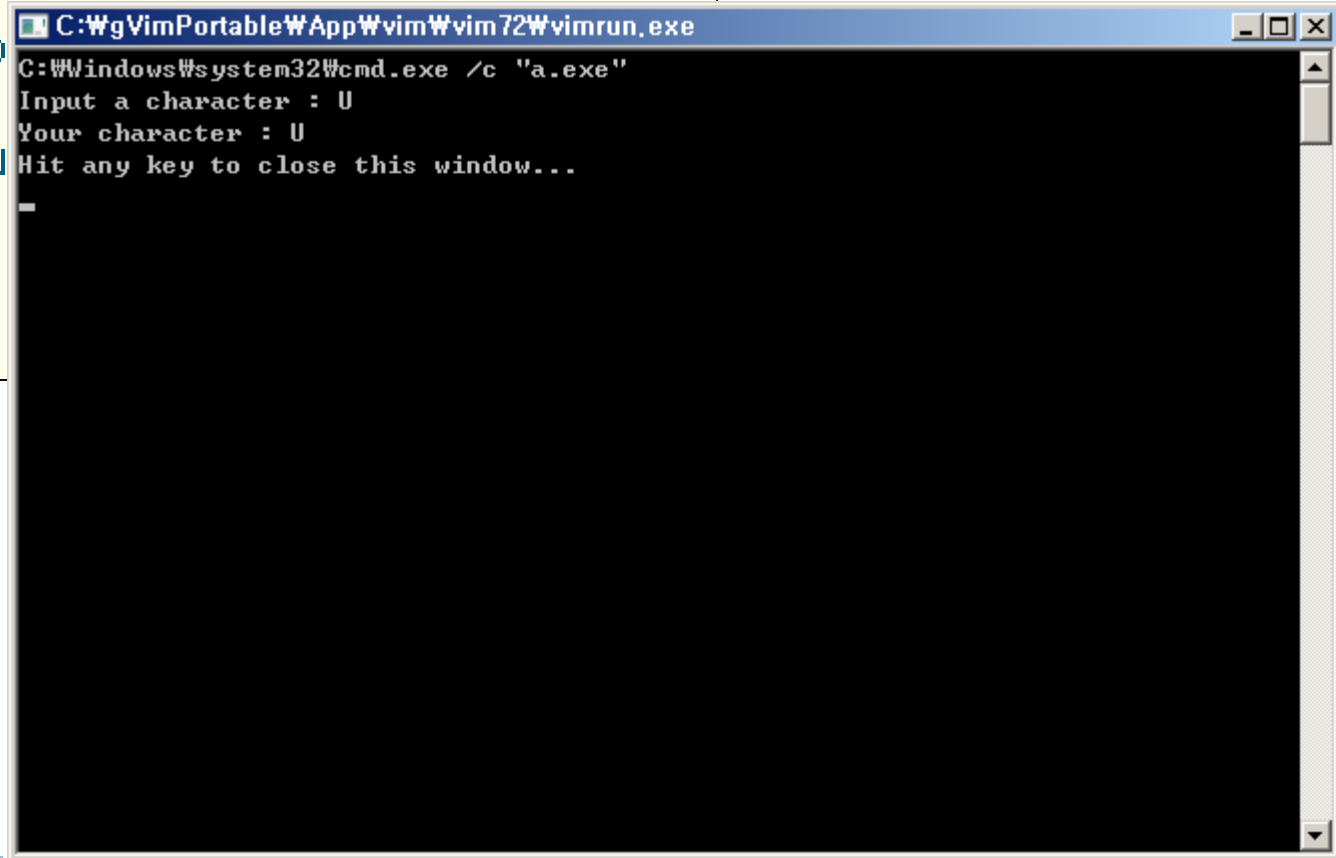




# Char Decl / Input

---

```
#include <stdio.h>
int main(){
    char x;
    printf("Input a character : ");
    scanf("%c", &x);
    printf("Your character : %c\n", x);
    //
    return 0;
}
```



```
C:\WgVimPortable\App\vim\vim72\vimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
Input a character : U
Your character : U
Hit any key to close this window...
```



# Char

---

- ▶ 숫자를 입력해도 이를 숫자가 아니라 문자로 인식
  - ▶ 무엇을 입력하던 문자로 인식함
  - ▶ 절대 혼동하지 말길!

'0' → 48



# Char

---

- ▶ ASCII Code 를 어떻게 하면 알아낼 수 있을까?

```
#include <stdio.h>
int main(){
    char x;
    printf("Input a character : ");
    scanf("%c",&x);
    printf("Your character : %c\n",x);
    printf("ASCII Code of your character is : ??????? );
    return 0;
}
```



# Char

---

- ▶ ASCII Code 를 어떻게 하면 알아낼 수 있을까?

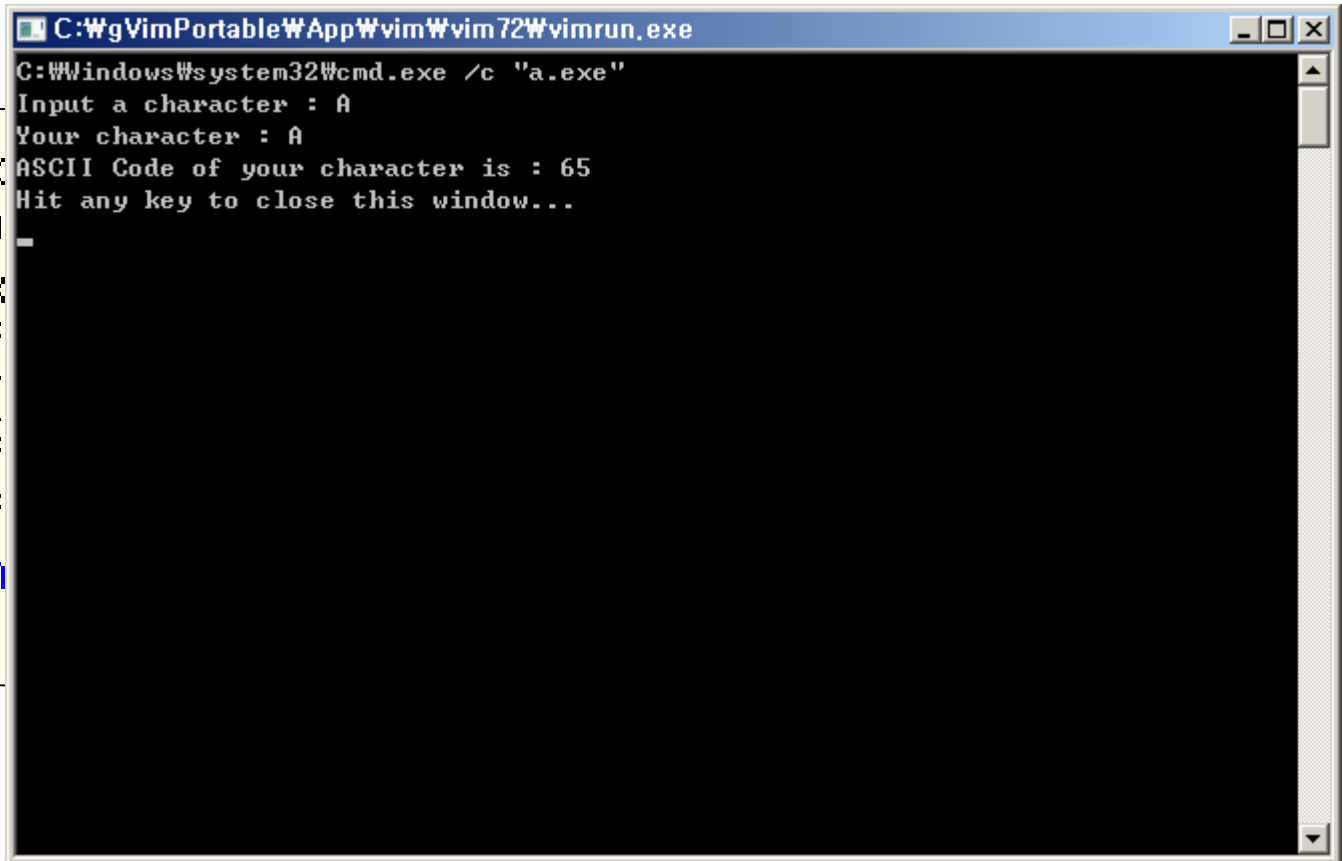
```
#include <stdio.h>
int main(){
    char x;
    printf("Input a character : ");
    scanf("%c",&x);
    printf("Your character : %c\n",x);
    printf("ASCII Code of your character is : %d",x);
    return 0;
}
```



# Char

- ▶ ASCII Code 를 어떻게 하면 알아낼 수 있을까?

```
#include <stdio.h>
int main()
{
    char x;
    printf("Input a character : ");
    scanf("%c", &x);
    printf("Your character : %c\n", x);
    printf("ASCII Code of your character is : %d\n", x);
    return 0;
}
```

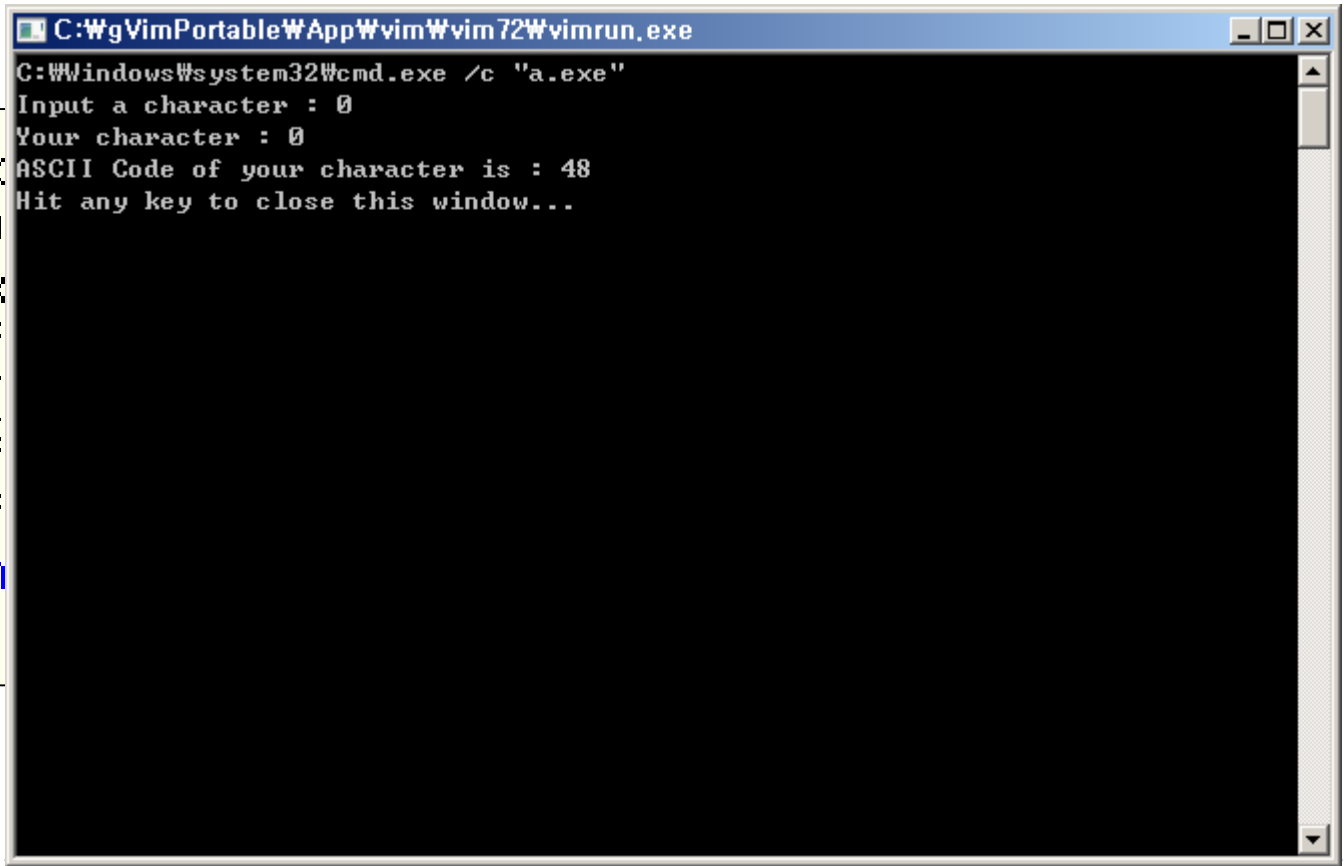


```
C:\WgVimPortable\WApp\vim\vim72\vimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
Input a character : A
Your character : A
ASCII Code of your character is : 65
Hit any key to close this window...
```

# Char

- ▶ ASCII Code 를 어떻게 하면 알아낼 수 있을까?

```
#include <stdio.h>
int main()
{
    char x;
    printf("Input a character : ");
    scanf("%c", &x);
    printf("Your character : %c\n", x);
    printf("ASCII Code of your character is : %d\n", x);
    return 0;
}
```



# Double

---

- ▶ 실수를 담는 변수
  - ▶ 매우 복잡한 저장 방법을 가짐 → 몰라도 됨
  - ▶ 범위가 꽤 큼. 이것도 복잡함 → 몰라도 됨



# Double Decl / Output

---

```
#include <stdio.h>
int main(){
    double x = 1.23;
    printf("%lf\n",x);
    return 0;
}
```

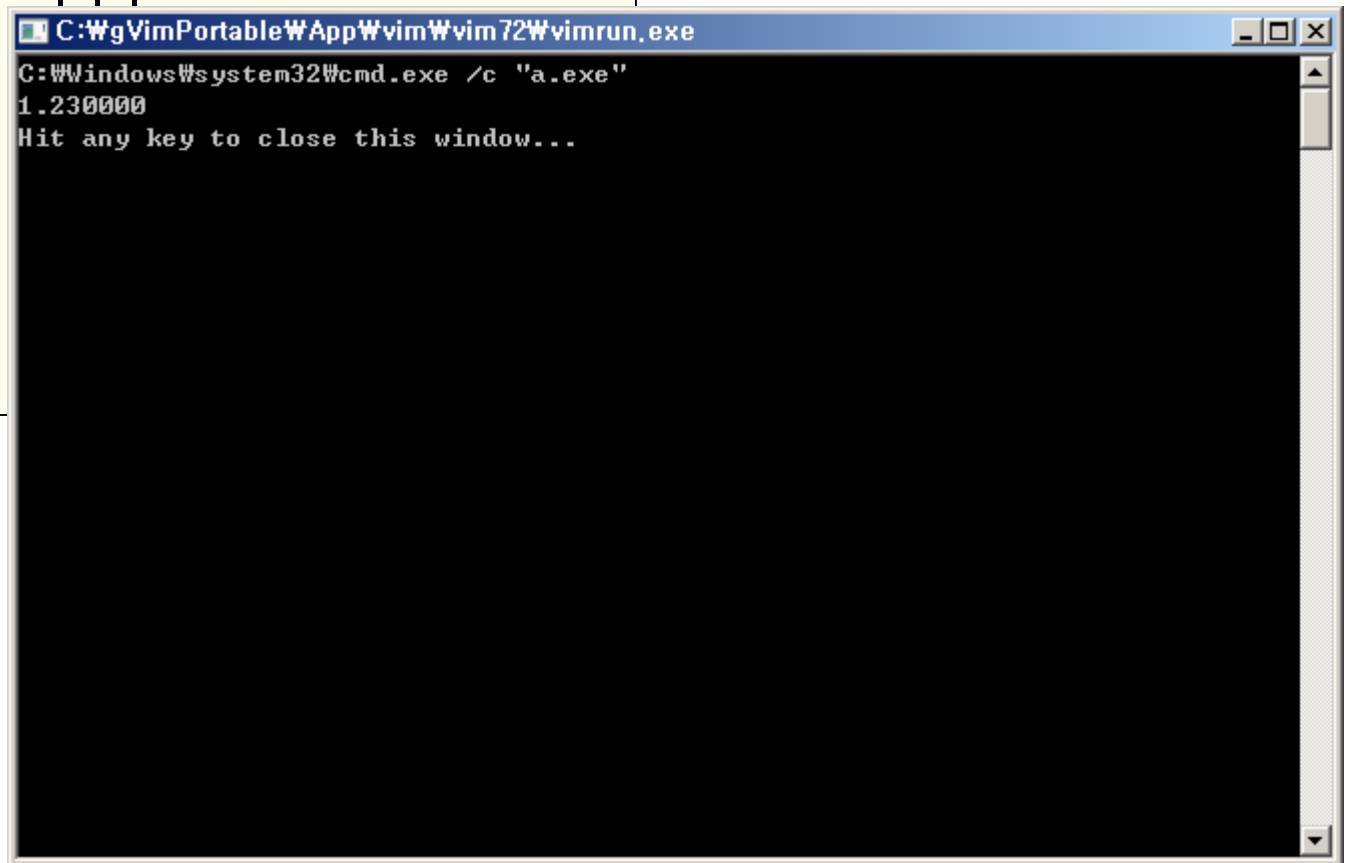




# Double Decl / Output

---

```
#include <stdio.h>
int main()
{
    double x = 1.23;
    printf("x = %f\n", x);
    return 0;
}
```



```
C:\WgVimPortable\WApp\vim\vim72\vimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
1.230000
Hit any key to close this window...
```



# Double Decl / Output

---

**%lf**



# Double Decl / Output

---

▶ `double x = 3.7245;`

`%10lf`

>



# Double Decl / Output

---

▶ `double x = 3.7245;`

`%10lf`

>



# Double Decl / Output

---

▶ `double x = 3.7245;`

`%10lf`

> 

		3	.	7	2	4	5	0	0
--	--	---	---	---	---	---	---	---	---



# Double Decl / Output

---

▶ `double x = 3.7245;`

`%10lf`

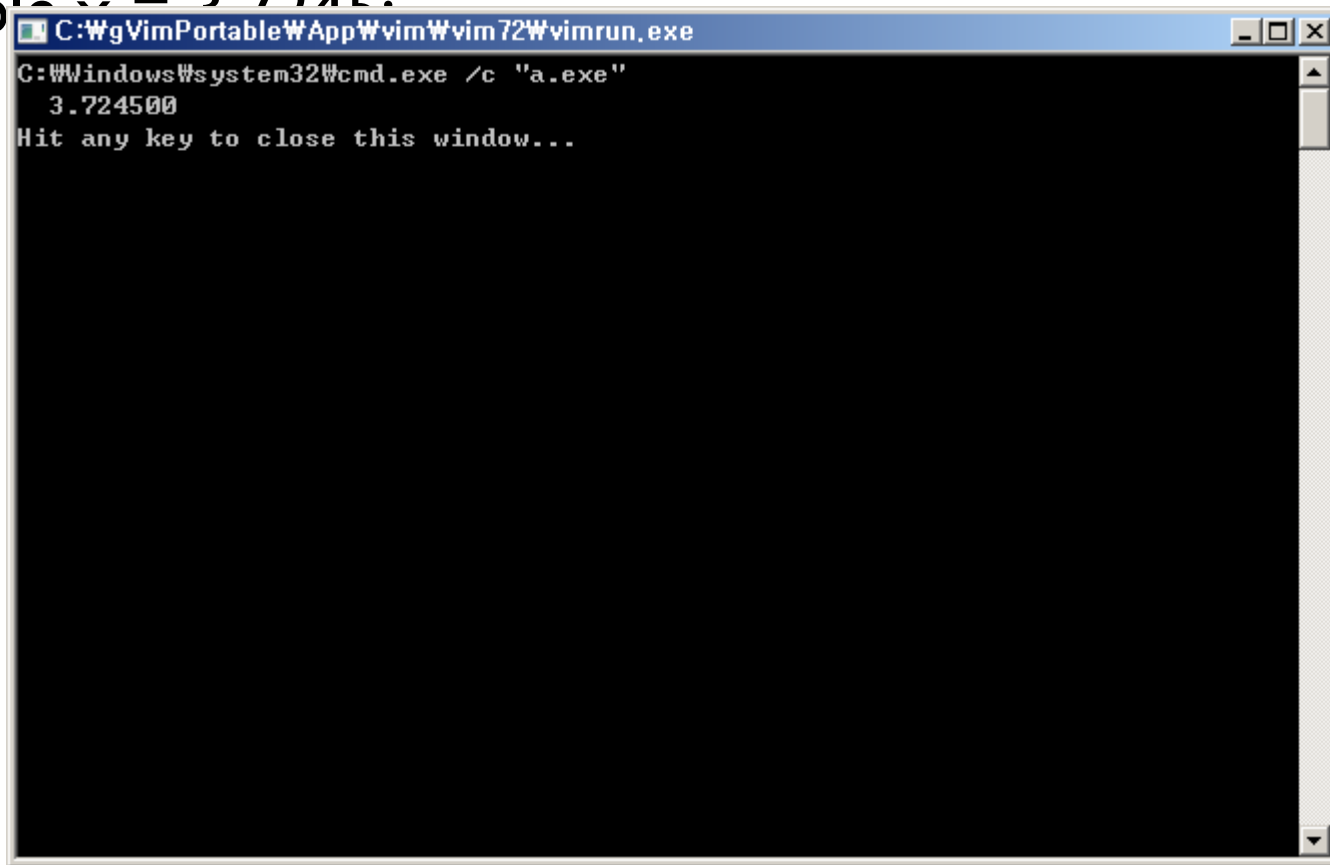
> 3 . 7 2 4 5 0 0



# Double Decl / Output

---

▶ `double x = 3.7245;`



The screenshot shows a Windows command prompt window with the following text:

```
C:\Windows\system32\cmd.exe /c "a.exe"  
3.724500  
Hit any key to close this window...
```

The window title bar reads: `C:\WgVimPortableWAppWvimWvim72Wvimrun.exe`. A small 'v' character is visible to the left of the window.



# Double Decl / Output

---

▶ `double x = 3.7246;`

`%.2lf`

>





# Double Decl / Output

---

▶ `double x = 3.7246;`

`%.2lf`

> 3 . 7 2



# Double Decl / Output

---

▶ `double x = 3.7246;`

`%.3lf`

>



# Double Decl / Output

---

▶ `double x = 3.7246;`

`%.3lf`

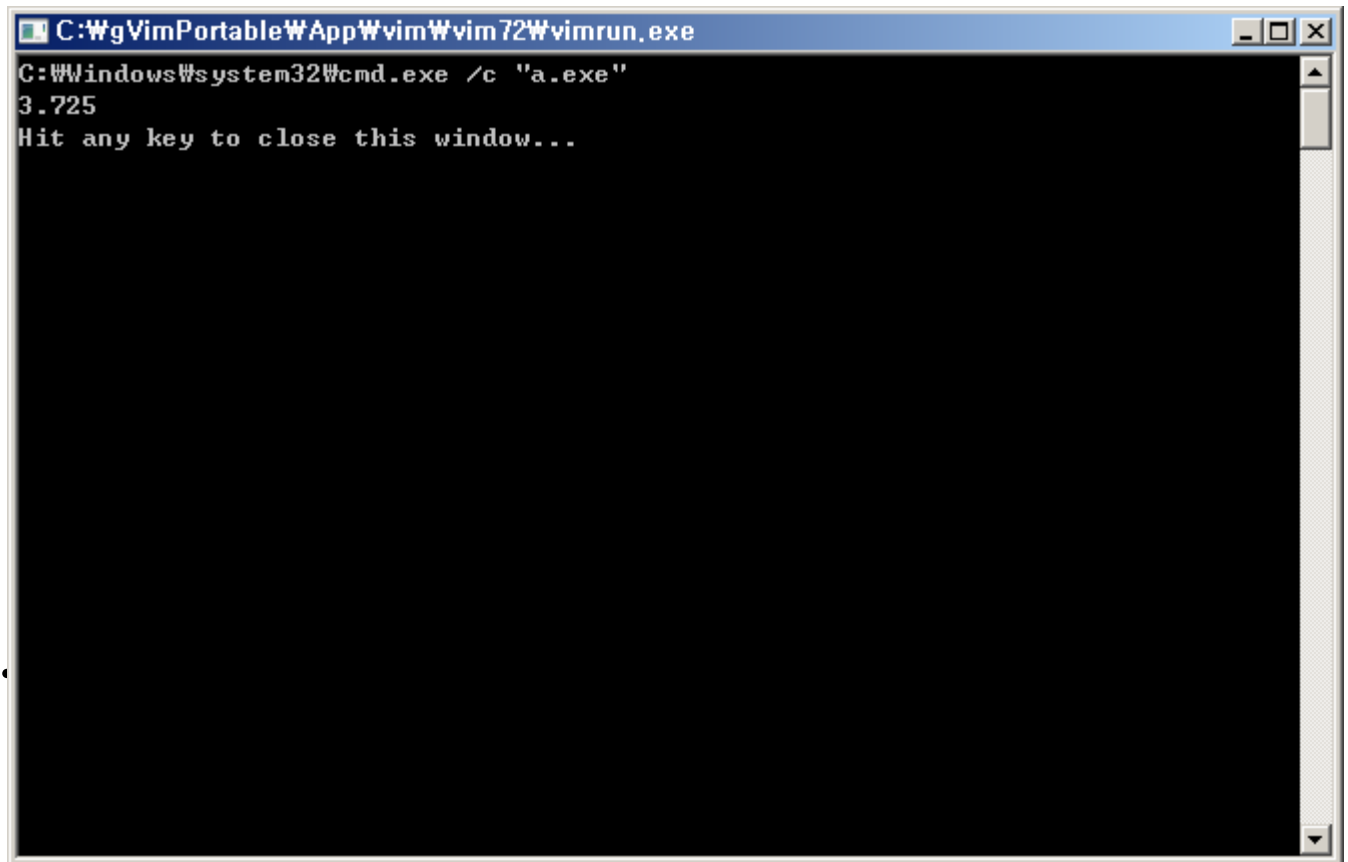
> 3 . 7 2 5



# Double Decl / Output

---

▶ `double x = 3.7246;`



```
C:\WgVimPortable\App\vim\vim72\vimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
3.725
Hit any key to close this window...
```

> 3



# Double Decl / Output

---

▶ `double x = 3.7246;`

`%10.2lf`

>



# Double Decl / Output

---

▶ `double x = 3.7246;`

`%10.2lf`

>



# Double Decl / Output

---

▶ `double x = 3.7246;`

`%10.2lf`

> 

						3	.	7	2
--	--	--	--	--	--	---	---	---	---



# Double Decl / Output

---

▶ `double x = 3.7246;`

**%10.2lf**

>

3 . 7 2

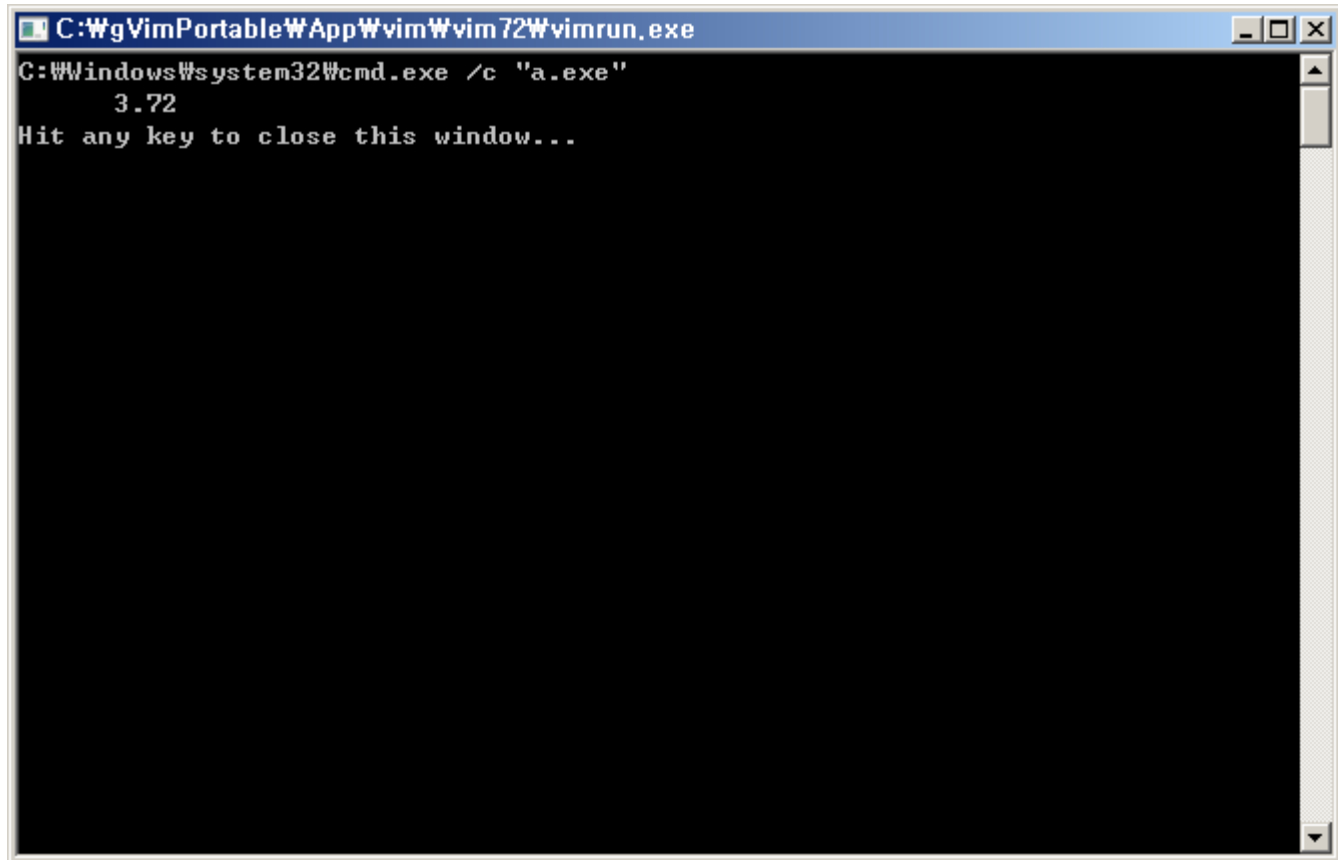




# Double Decl / Output

---

- ▶ `double x = 3.7246;`



A screenshot of a Windows command prompt window. The title bar reads "C:\WgVimPortable\App\vim\vim72\vimrun.exe". The command prompt shows the following text:

```
C:\Windows\system32\cmd.exe /c "a.exe"  
3.72  
Hit any key to close this window...
```

>



# Double Input

---

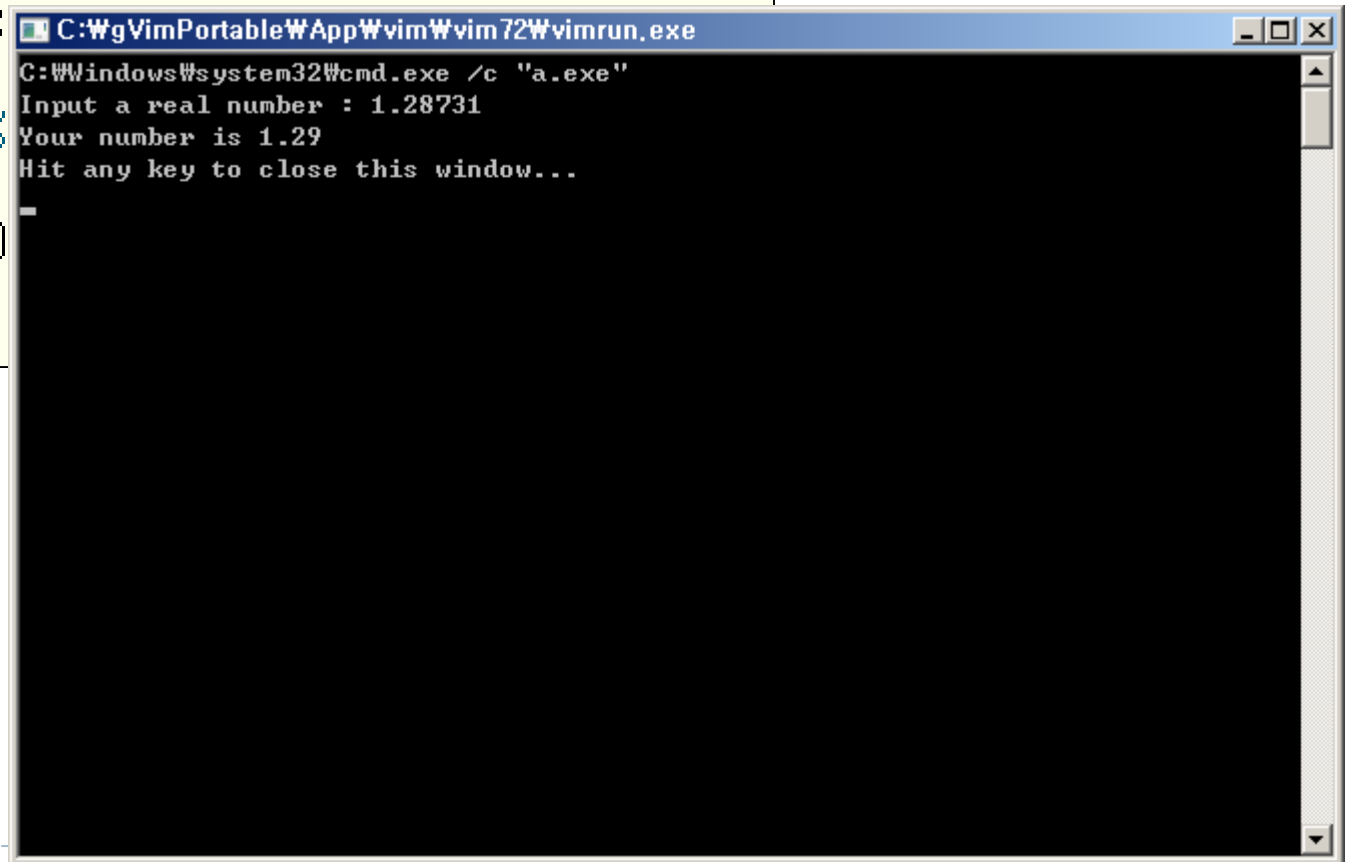
```
#include <stdio.h>
int main(){
    double x;
    printf("Input a real number : ");
    scanf("%lf",&x);
    printf("Your number is %.2lf#\n",x);
    return 0;
}
```



# Double Input

---

```
#include <stdio.h>
int main(){
    double x;
    printf("Input a real number : ");
    scanf("%lf", &x);
    printf("Your number is %.2f\n", x);
    return 0;
}
```



```
C:\WgVimPortableWAppWvimWvim72Wvimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
Input a real number : 1.28731
Your number is 1.29
Hit any key to close this window...
-
```

# Double

---

- ▶ 세 개의 정수를 입력 받아 평균을 구해보자!
  - ▶ 어떻게 짤까?



# Double

---

- ▶ 세 개의 정수를 입력 받아 평균을 구해보자!
  - ▶ 어떻게 짤까? → **Algorithm Design!**



# Double

---

- ▶ 세 개의 정수를 입력 받아 평균을 구해보자!
  - ▶ 어떻게 짤까? → **Algorithm Design!**
    1. 우선 세 개의 숫자를 입력 받고
    2. 그 세 숫자의 합을 구한 뒤
    3. 이를 3으로 나누어서 평균을 구하자!



# Double

---

```
#include <stdio.h>
int main(){
    int a, b, c;
    int sum = 0;
    double average;
    printf("Input three numbers : ");
    scanf("%d %d %d",&a, &b, &c);
    sum = a + b + c;
    average = sum / 3;
    printf("Sum is %d#\n",sum);
    printf("Average is %.2lf#\n",average);
    return 0;
}
```



# Double

```
#include <stdio.h>
```

```
int main(){
```

```
    int a, b, c;
```

```
    int sum;
```

```
    double average;
```

```
    printf("Input three numbers : ");
```

```
    scanf("%d %d %d", &a, &b, &c);
```

```
    sum = a + b + c;
```

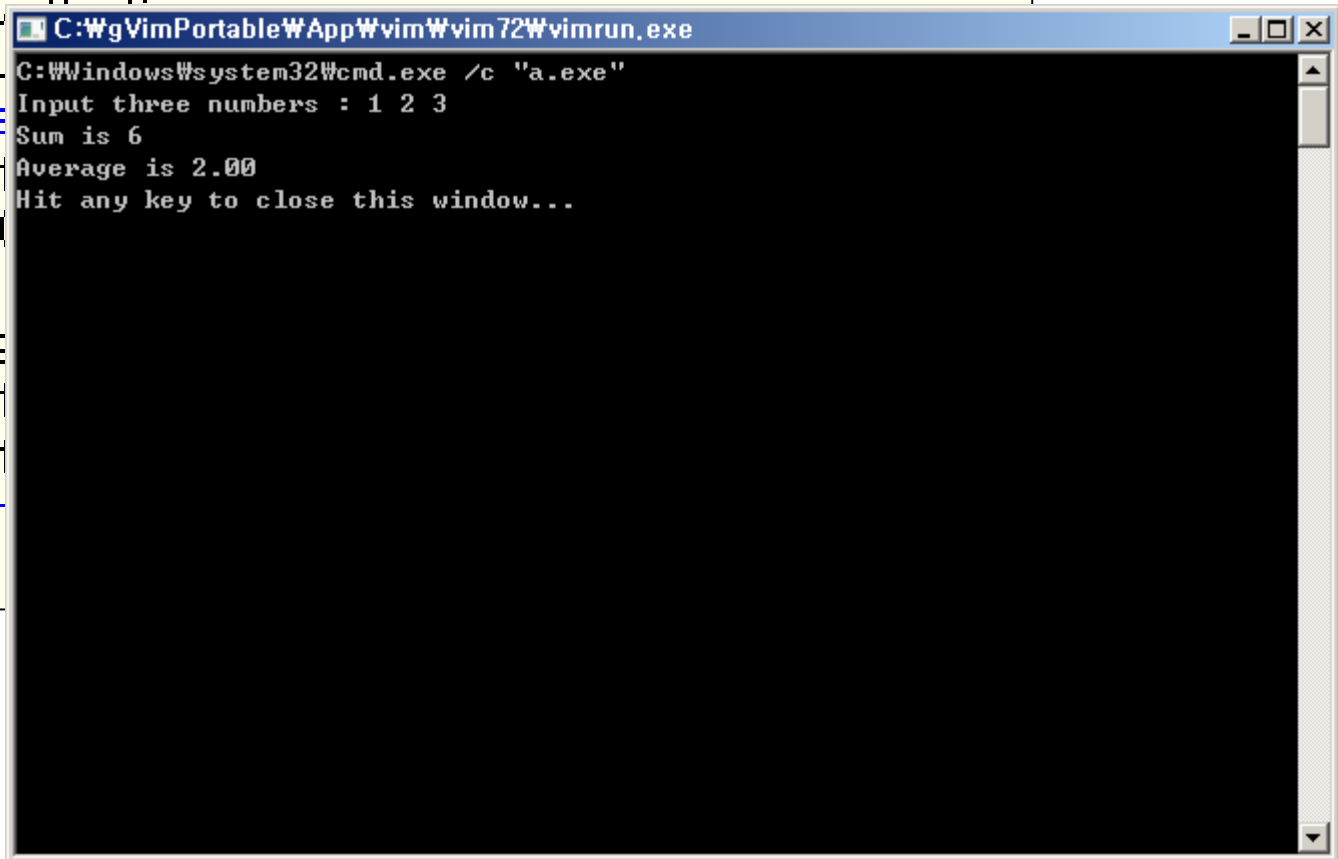
```
    average = sum / 3;
```

```
    printf("Sum is %d\n", sum);
```

```
    printf("Average is %.2f\n", average);
```

```
    return 0;
```

```
}
```



```
C:\WgVimPortable\App\vim\vim72\vimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
Input three numbers : 1 2 3
Sum is 6
Average is 2.00
Hit any key to close this window...
```





# Double

```
#include <stdio.h>
```

```
int main(){
```

```
    int
```

```
    int
```

```
    doub
```

```
    prin
```

```
    scan
```

```
    sum
```

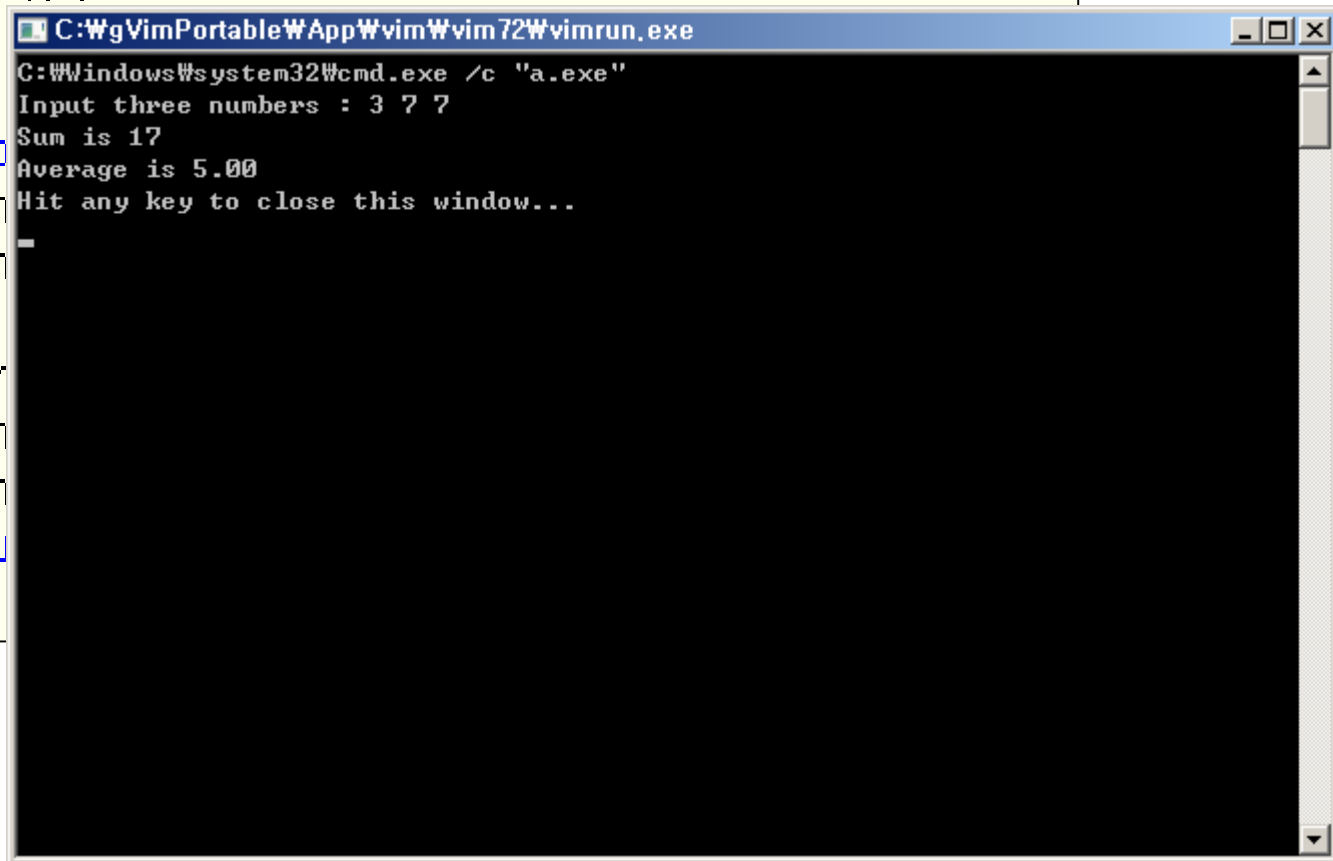
```
    aver
```

```
    prin
```

```
    prin
```

```
    retu
```

```
}
```



```
C:\WgVimPortable\App\vim\vim72\vimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
Input three numbers : 3 7 7
Sum is 17
Average is 5.00
Hit any key to close this window...
```



# Double

```
#include <stdio.h>
```

```
int main(){
```

```
    int
```

```
    int
```

```
    doub
```

```
    prin
```

```
    scan
```

```
    sum
```

```
    aver
```

```
    prin
```

```
    prin
```

```
    retu
```

```
}
```

```
C:\WgVimPortable\App\vim\vim72\vimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
Input three numbers : 3 7 7
Sum is 17
Average is 5.00
Hit any key to close this window...
```

Wrong Answer!

# Casting

---

- ▶ int 끼리의 연산은 int형으로 반환
  - ▶ 같은 Type끼리의 연산은 그 Type의 연산으로 취급
    - `double x = 5 / 3;`
    - int 나누기 int는 int형으로 연산. 따라서 x는 1.00이 됨
  - ▶ 다른 Type끼리의 연산의 경우 더 큰 범위의 연산으로 취급
    - `double x = 5 / 3.00`
    - int 나누기 double은 double형으로 연산. 따라서 x는 1.67이 됨
- ▶ Type을 알맞게 바꾸어 주어야 함
  - ▶ Casting ( Type Conversion )
  - ▶ 앞에 바꾸고자 하는 Type으로 묶어줌



# Casting

```
#include <stdio.h>
int main(){
    int a, b, c;
    int sum = 0;
    double average;
    printf("Input three numbers : ");
    scanf("%d %d %d",&a, &b, &c);
    sum = a + b + c;
    average = sum / 3;
    printf("Sum is %d\n",sum);
    printf("Average is %.2lf\n",average);
    return 0;
}
```

Casting이 없어 int끼리의 연산으로 취급  
→ average에 double 형의 값이 들어가는 것이  
아닌 int형의 값이 들어감



# Casting

---

```
#include <stdio.h>
int main(){
    int a, b, c;
    int sum = 0;
    double average;
    printf("Input three numbers : ");
    scanf("%d %d %d",&a, &b, &c);
    sum = a + b + c;
    average = (double)sum / 3;
    printf("Sum is %d\n",sum);
    printf("Average is %.2lf\n",average);
    return 0;
}
```



# Casting

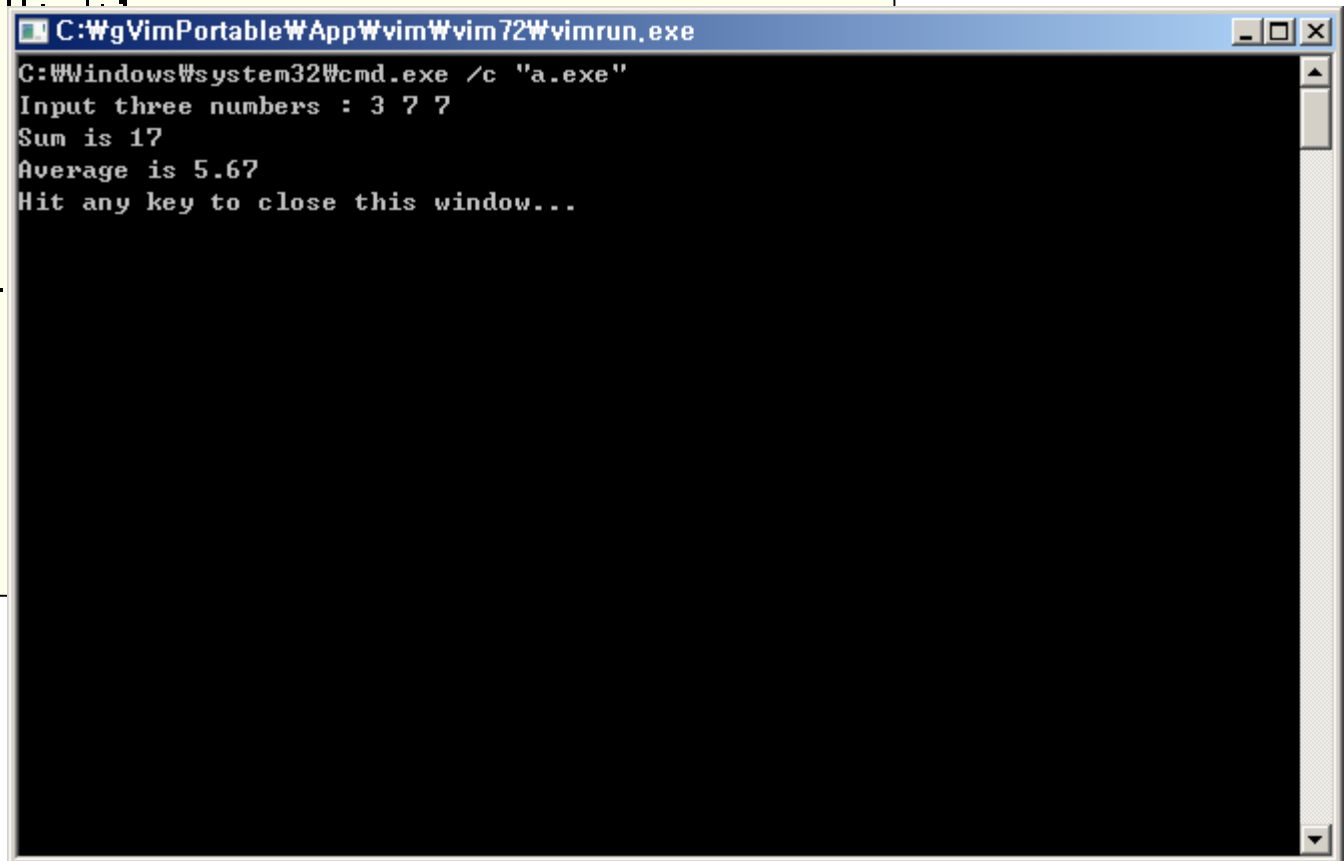
---

```
#include <stdio.h>
int main(){
    int a, b, c;
    int sum = 0;
    double average;
    printf("Input three numbers : ");
    scanf("%d %d %d",&a, &b, &c);
    sum = a + b + c;
    average = (double)sum / 3;
    printf("Sum is %d\n",sum);
    printf("Average is %.2lf\n",average);
    return 0;
}
```



# Casting

```
#include <stdio.h>
int main(){
    int a, b, c;
    int sum;
    double
    printf(
    scanf("
    sum = a
    average
    printf(
    printf(
    return
}
```



```
C:\WgVimPortable\App\vim\vim72\vimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
Input three numbers : 3 7 7
Sum is 17
Average is 5.67
Hit any key to close this window...
```

# Structure

---

- ▶ 이거 말고 다른 Type은 없나?
  - ▶ 새로운 Type을 정의할 수 있음
    - 점  $(x, y)$ 의 정보를 저장하는 변수
    - 학번과 그 학번의 평균 학점(ㅋㅋ)을 저장하는 변수
  - ▶ **Structure** 가 있음
- ▶ Structure
  - ▶ Type을 새롭게 정의함

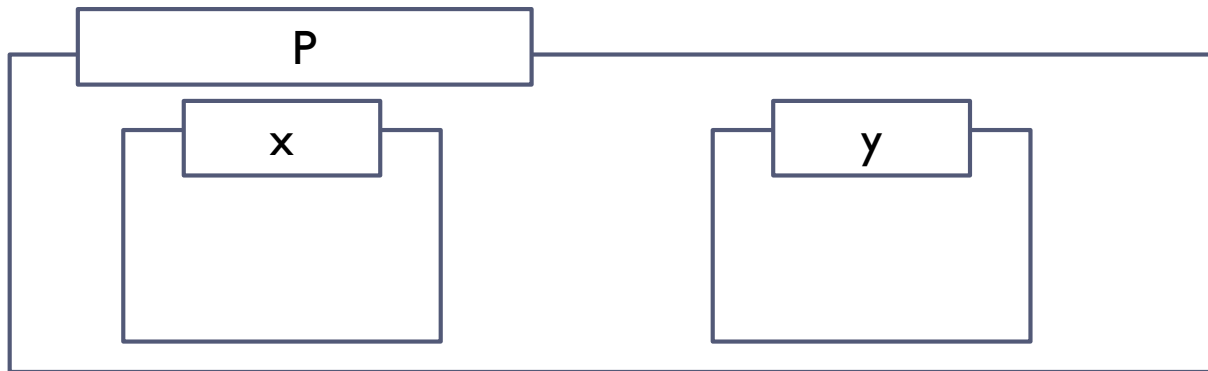




# Structure

---

- ▶ 이거 말고 다른 Type은 없나?
  - ▶ 새로운 Type을 정의할 수 있음
    - 점  $(x, y)$ 의 정보를 저장하는 변수
    - 학번과 그 학번의 평균 학점(ㅋㅋ)을 저장하는 변수
  - ▶ **Structure** 가 있음
- ▶ Structure
  - ▶ Type을 새롭게 정의함



# Structure

---

```
#include <stdio.h>
struct p{
    int x;
    int y;
};
int main(){
    struct p p1, p2;
    p1.x = 3;
    p1.y = 4; // information of p1
    p2.x = 2;
    p2.y = 3; // information of p2

    printf("p1 = (%d, %d)\n",p1.x, p1.y);
    printf("p2 = (%d, %d)\n",p2.x, p2.y);
    return 0;
}
```

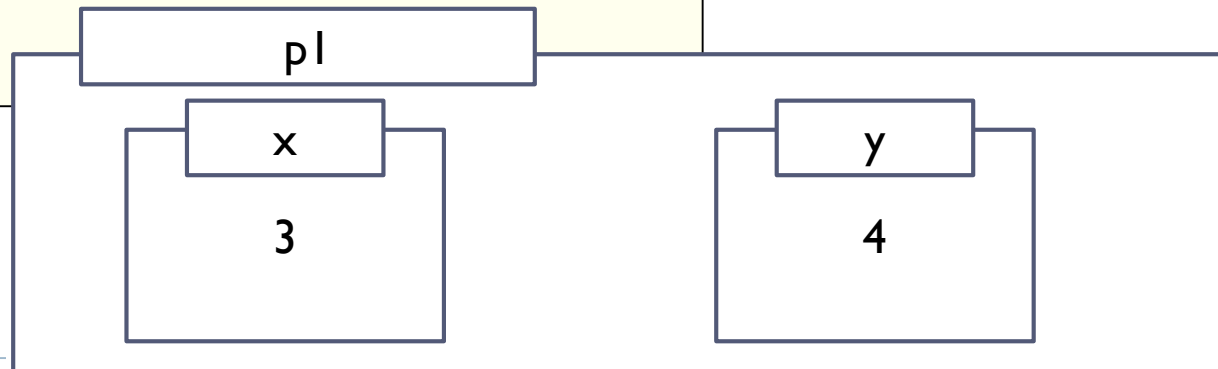


# Structure

```
#include <stdio.h>
struct p{
    int x;
    int y;
};
int main(){
    Type struct p p1, p2;
    p1.x = 3;
    p1.y = 4; // information of p1
    p2.x = 2;
    p2.y = 3; // information of p2

    printf("p1 = (%d, %d)\n", p1.x, p1.y);
    printf("p2 = (%d, %d)\n", p2.x, p2.y);
    return 0;
}
```

Type 정의



# Structure

```
#include <stdio.h>
struct p{
    int x;
    int y;
};
int main(){
    struct p p1, p2;
    p1.x = 3;
    p1.y = 4; // inf
    p2.x = 2;
    p2.y = 3; // inf

    printf("p1 = (%d
    printf("p2 = (%d
    return 0;
}
```

C:\WgVimPortableWAppWvimWvim72Wvimrun.exe

```
C:\Windows\system32\cmd.exe /c "a.exe"
p1 = <3, 4>
p2 = <2, 3>
Hit any key to close this window...
```

# Structure

---

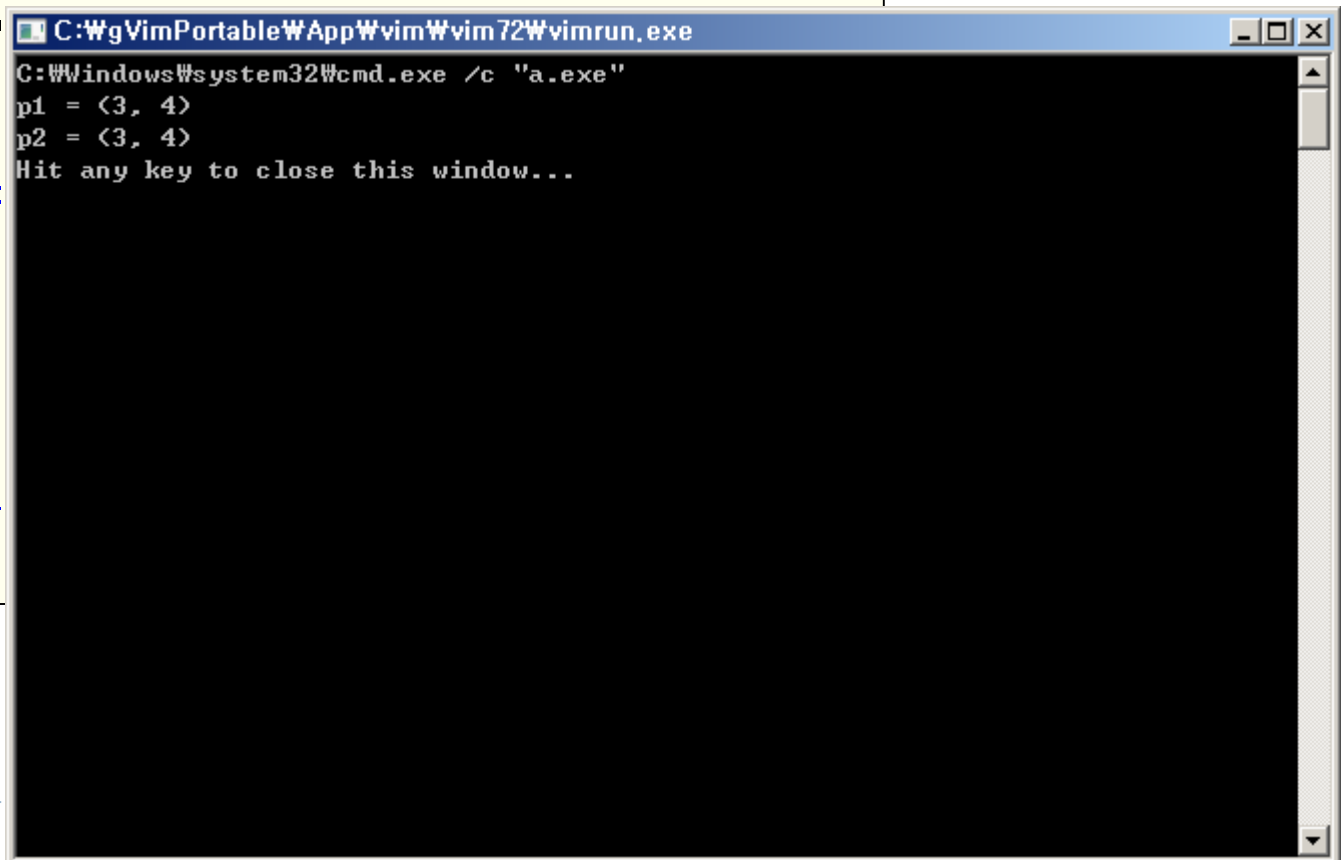
- ▶ Structure 에는 Operator가 없음
  - ▶ **Assignment operator**는 제외
  - ▶ 새로운 Type을 정의했으니 연산이 없는 것이 당연
  - ▶  $(x1, y1) + (x2, y2) = ?$
  - ▶ 이것 또한 정의해줄 수 있으나 몰라도 됨
- ▶ **Assignment operator**
  - ▶ Member-wise copy



# Structure

---

```
#include <stdio.h>
struct p{
    int x;
    int y;
};
int main(
    struc
    p1.x
    p1.y
    p2 =
    print
    print
    retur
}
```



```
C:\WgVimPortable\App\vim\vim72\vimrun.exe
C:\Windows\system32\cmd.exe /c "a.exe"
p1 = (3, 4)
p2 = (3, 4)
Hit any key to close this window...
```

# Structure

---

## ▶ Why use it?

- ▶ 여러 가지의 정보를 한꺼번에 갖고 다녀야 할 때
  - Point (x, y)
  - 학생의 정보 (학번, 이름, **학점**)
- ▶ 지금은 별 필요성을 못 느낄 수 있지만 Algorithm가면 빠져리게...



# Contents

---

