

start

Today

A little review

pointers

strings

I/O

CPP: A language within!

Finally, hello world!

Beyond int: floats, doubles

Compound data types:

struct

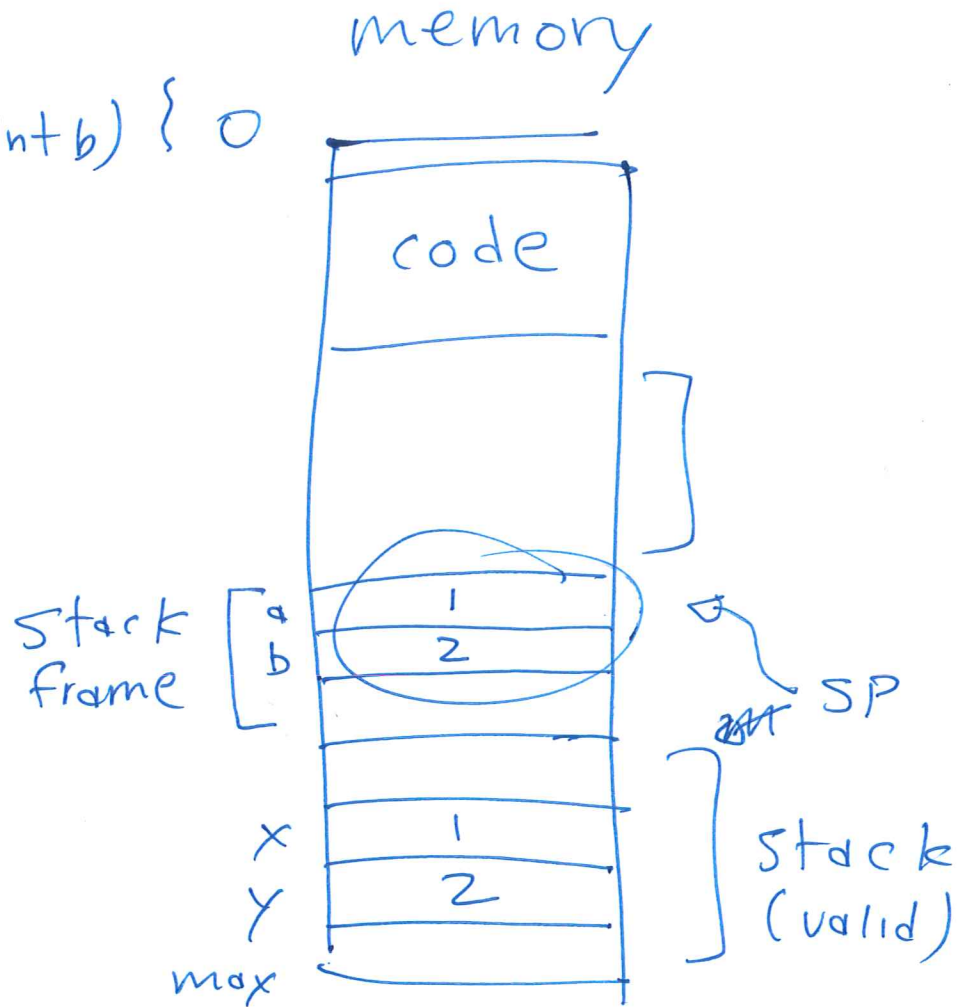
The Third Type of Memory

The C Library:

A quick tour

①

```
void swap  
  (inta, intb) {  
    ==  
  }  
}
```



swap(x, y)

①

```
void swap-int (int* a, int* b) {  
    int tmp = *a;  
    *a = *b;  
    *b = tmp;  
}
```

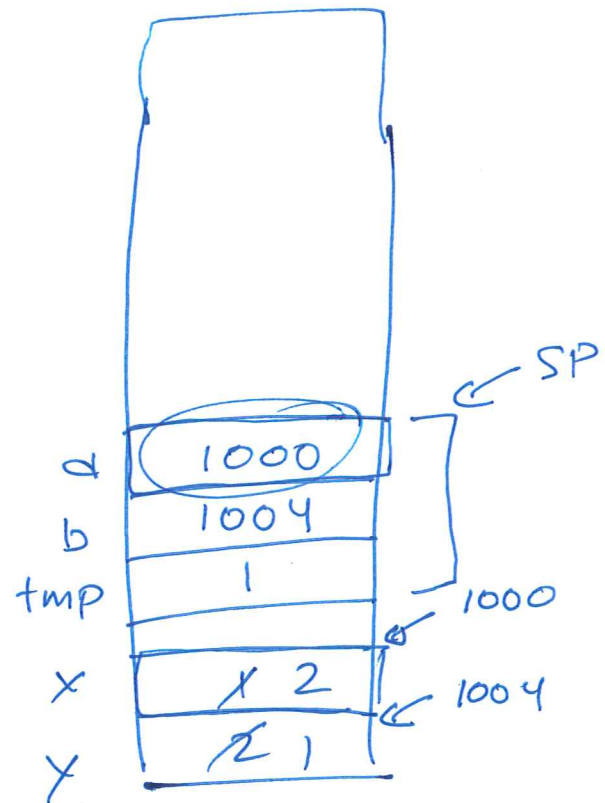
```
int x, y;
```

```
x = 1;
```

```
y = 2;
```

```
swap-int (x, y);
```

function
call
passes by
value
(copy)



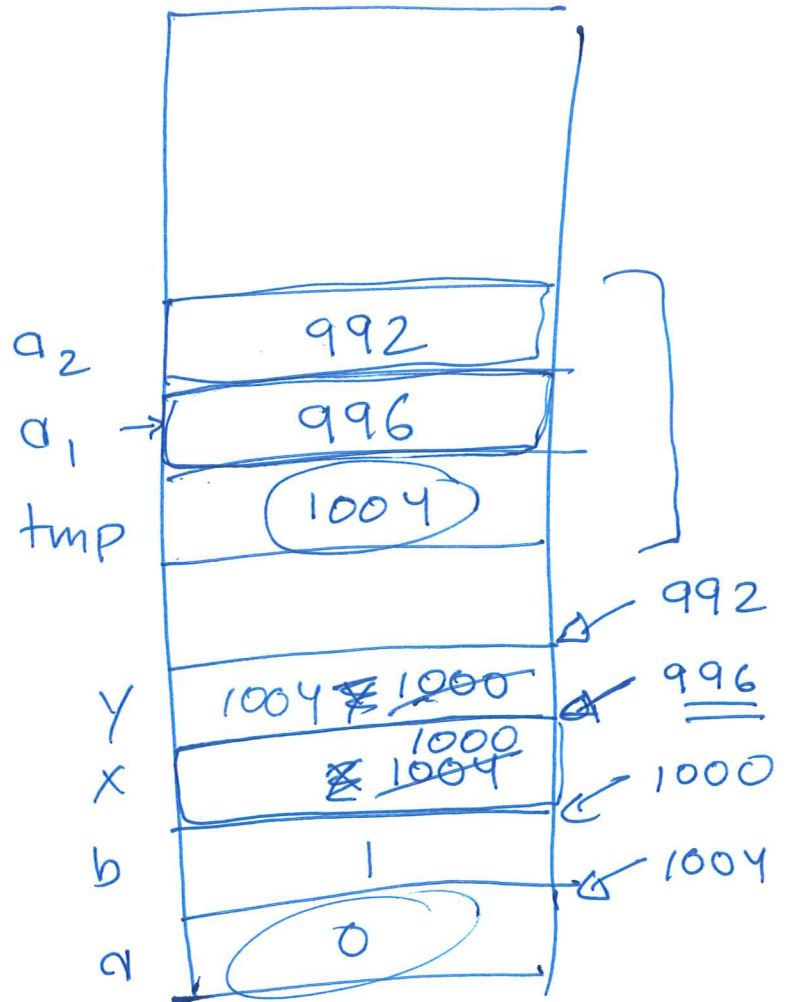
2

```
swap-int_ptr(int**a, int**a2) {  
    int* tmp;
```

```
    tmp = *a1;
```

```
    *a1 = *a2;
```

```
    *a2 = tmp;
```



```
x = 3 a;
```

```
y = 3 b;
```

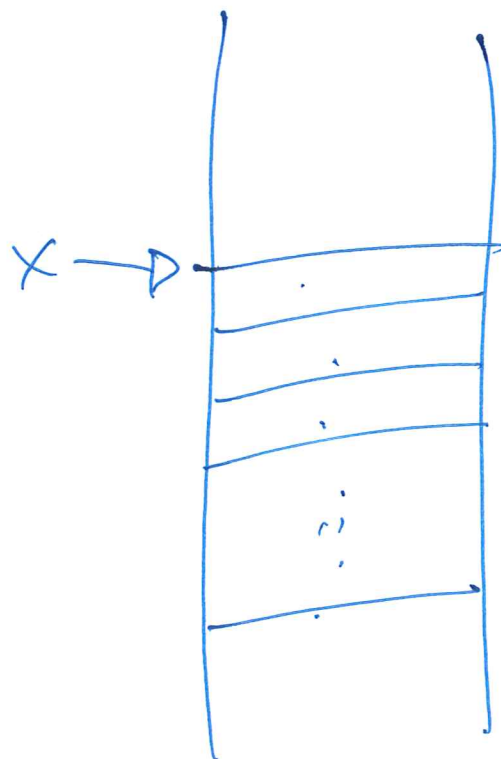
```
swap-int_ptr(3 x, 3 y);
```

③

$a[i]$

$\Rightarrow * (a + i)$

$\text{array}[i] += 1$



Unix: input/output

(4)

terminal \Rightarrow type,
screen

stdin \Rightarrow type: (FILE*)

Standard input

\Rightarrow read to
get input
from user

stdout \Rightarrow

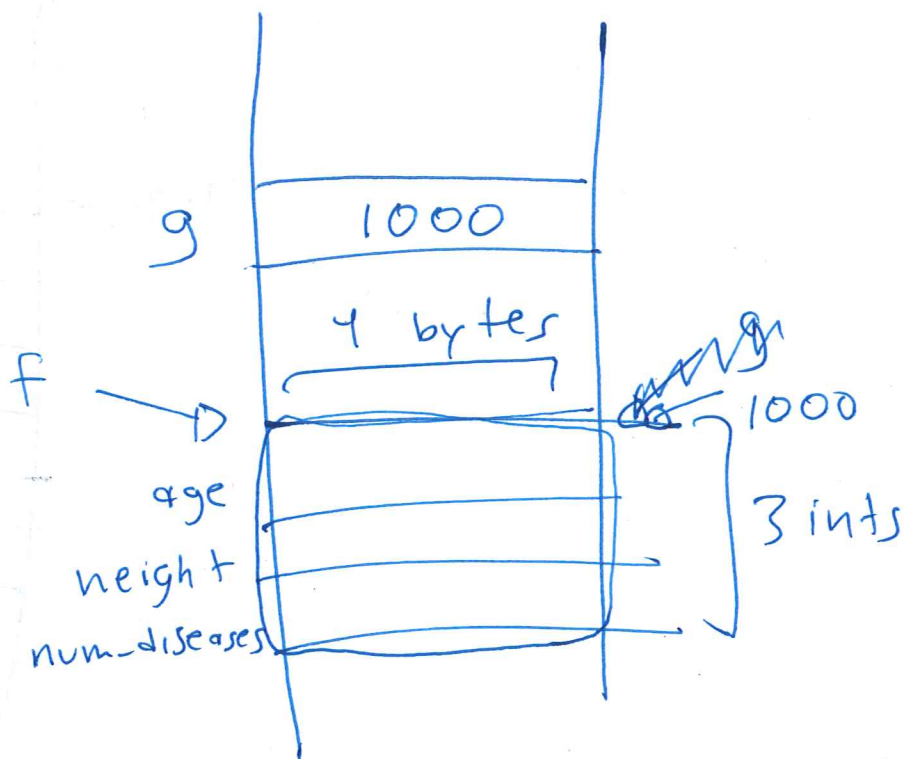
Standard output

file
streams

\Rightarrow write to
show something
on screen

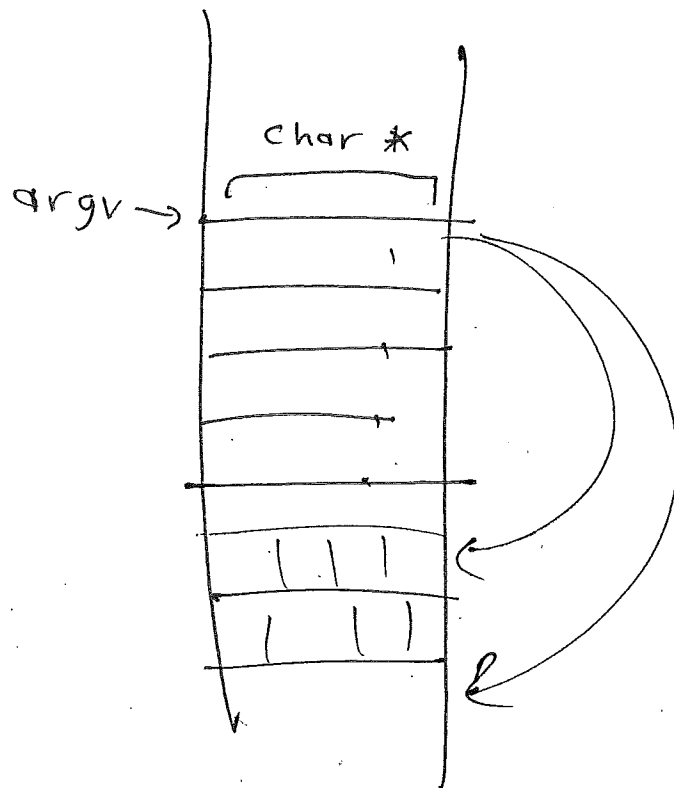
[third: output \Rightarrow stderr]

6



$$g = 3f$$

char* argv[]



C

control

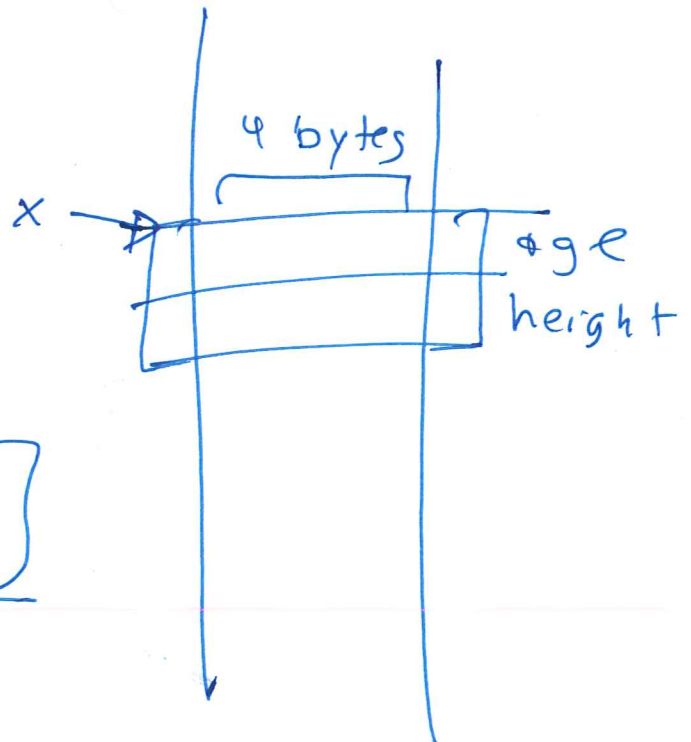
⑦

layout in
memory

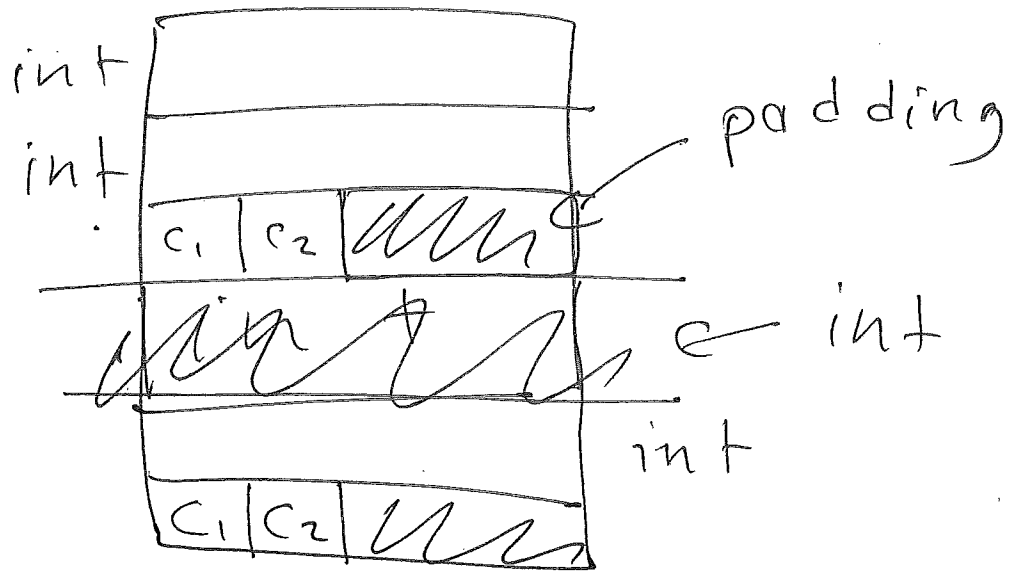
```
struct hr {  
    int age;  
    int height;  
};
```

```
struct hr x;
```

```
sizeof(x)  
=> 8 bytes
```



8



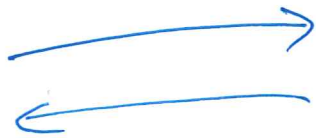
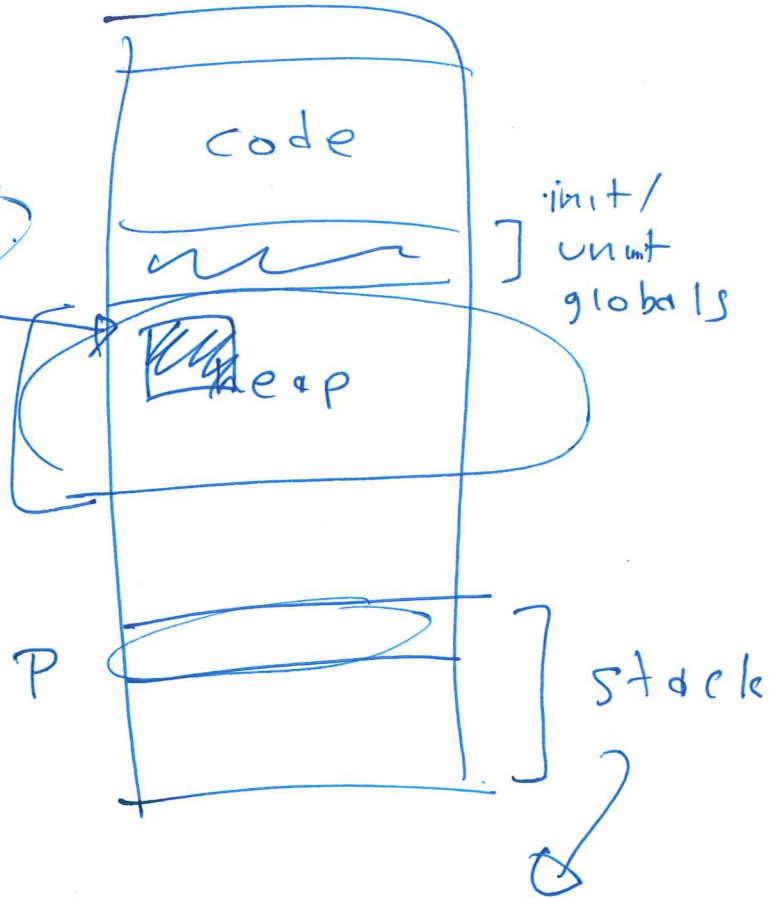
```
struct info {  
    int  
    int  
    char  
    char  
}
```

(Memory)
Address Space ⑨

8
8

memory
that is
more flexible
than
stack

0x78-----



automatic
⇒ C alloc/
dealloc
for you

heap:
allocate/deallocate
ourselves

alloc ⇒ malloc (size (in bytes))
dealloc ⇒ free (ptr to malloc'd space)

```
char buffer [MAX]
```

```
while (fgets(buffer, MAX, stdin) != NULL) {
```

```
    reverse(index++) = buffer;
```

```
}
```

