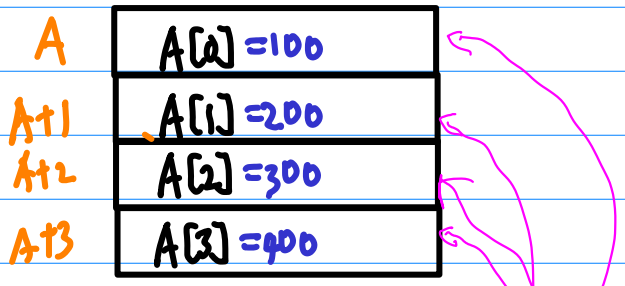
C Programming
Day13.B

2017.10.24

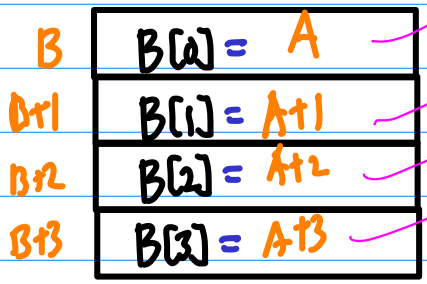
Array of Pointers,
main's arguments
and return value

an array of pointers

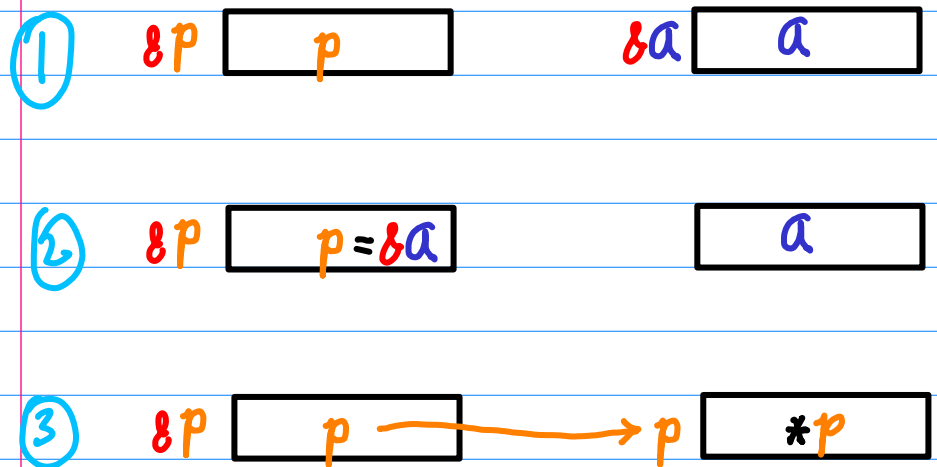
```
int  A[4];
```

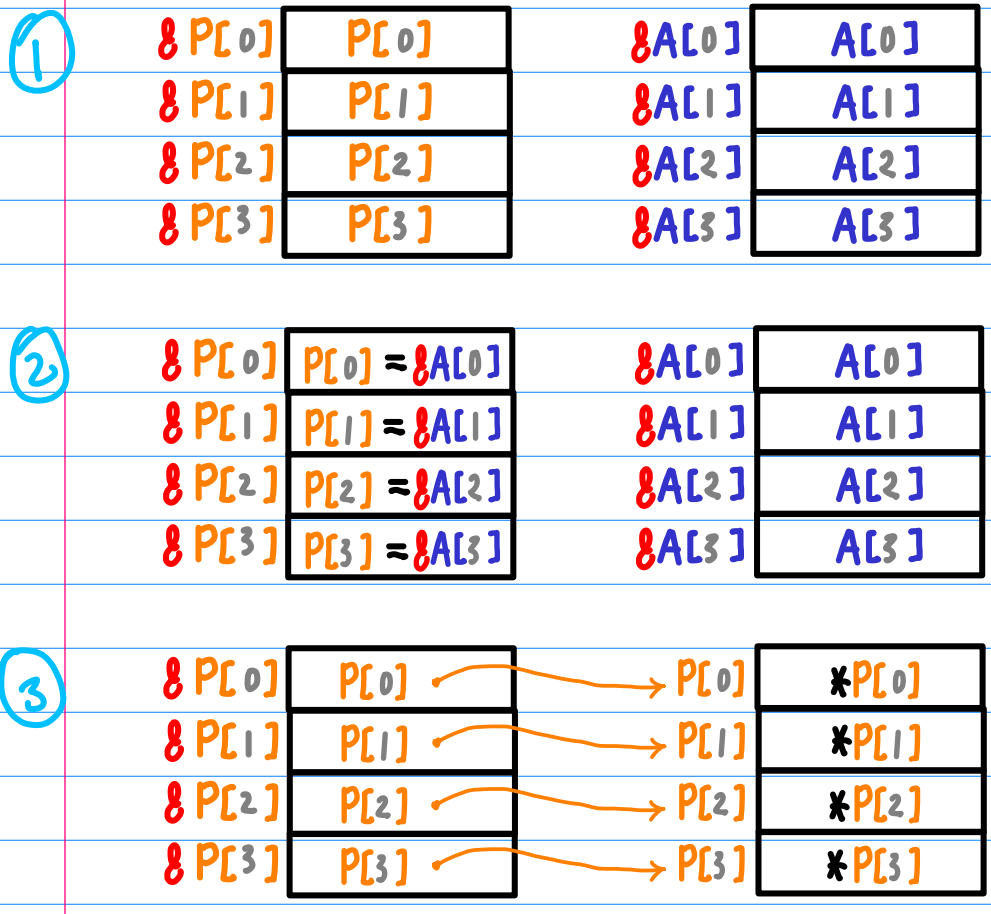



```
int *B[4];
```



```c
#include <stdio.h>

int main(void) {
  int a     =   100;
  int A[4]  = { 100, 200, 300, 400 };

  int *p    =   &a;
  int *P[4] = { &A[0], &A[1], &A[2], &A[3] };

  int i;

  printf("--------|---------------\n");
  printf("&a= %p  ", &a);
  printf(" a= %d\n",  a);

  printf("-----------------------\n");
  printf("&p= %p  ", &p);
  printf(" p= %p  ",  p);
  printf("*p= %d\n", *p);

  printf("-----------------------\n");
  for (i=0; i<4; ++i) {
    printf("&A[%d]= %p  ", i, &A[i]);
    printf(" A[%d]= %d\n", i,  A[i]);
  }

  printf("-----------------------\n");
  for (i=0; i<4; ++i) {
    printf("&P[%d]= %p  ", i, &P[i]);
    printf(" P[%d]= %p  ", i,  P[i]);
    printf("*P[%d]= %d\n", i, *P[i]);
  }
}
```


1
&p
p
&a
a
2
&p
p=&a
a
3
&p
p
p
*p


1
&P[0]
P[0]
&P[1]
P[1]
&P[2]
P[2]
&P[3]
P[3]
&A[0]
A[0]
&A[1]
A[1]
&A[2]
A[2]
&A[3]
A[3]
2
&P[0]
P[0]=&A[0]
&P[1]
P[1]=&A[1]
&P[2]
P[2]=&A[2]
&P[3]
P[3]=&A[3]
&A[0]
A[0]
&A[1]
A[1]
&A[2]
A[2]
&A[3]
A[3]
3
&P[0]
P[0]
&P[1]
P[1]
&P[2]
P[2]
&P[3]
P[3]
P[0]
*P[0]
P[1]
*P[1]
P[2]
*P[2]
P[3]
*P[3]

```
----------------------
&a= 0x7ffce6627990   a= 100
----------------------
&p= 0x7ffce6627998   p= 0x7ffce6627990  *p= 100
----------------------
&A[0]= 0x7ffce66279a0   A[0]= 100
&A[1]= 0x7ffce66279a4   A[1]= 200
&A[2]= 0x7ffce66279a8   A[2]= 300
&A[3]= 0x7ffce66279ac   A[3]= 400
----------------------
&P[0]= 0x7ffce66279b0   P[0]= 0x7ffce66279a0  *P[0]= 100
&P[1]= 0x7ffce66279b8   P[1]= 0x7ffce66279a4  *P[1]= 200
&P[2]= 0x7ffce66279c0   P[2]= 0x7ffce66279a8  *P[2]= 300
&P[3]= 0x7ffce66279c8   P[3]= 0x7ffce66279ac  *P[3]= 400
```

$$\&A[0] \equiv \&(A[0]) \equiv (A+0)$$

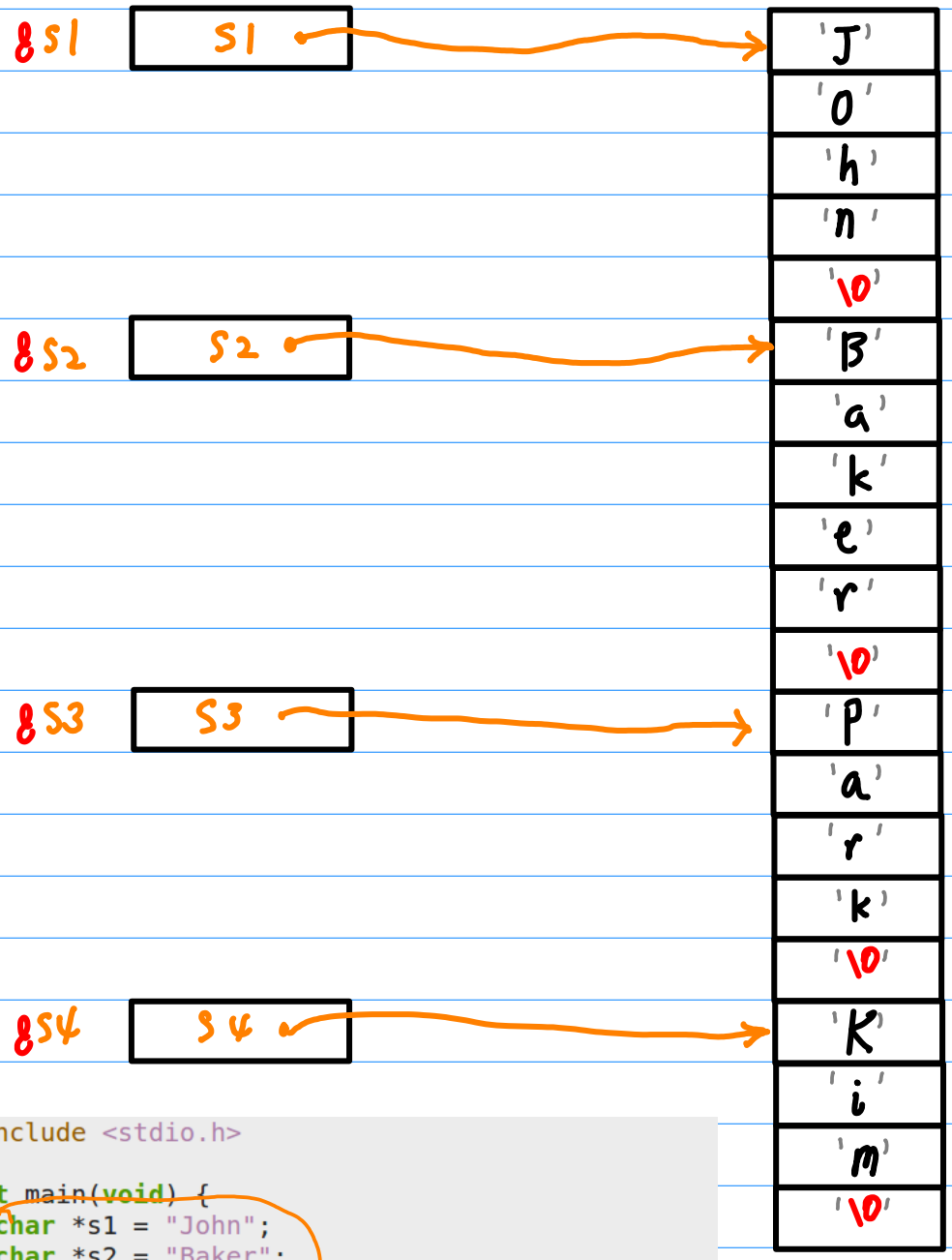
$$A[0] \equiv *(A+0)$$

&A[0]	A[0]	(A+0)	*(A+0)
&A[1]	A[1]	(A+1)	*(A+1)
&A[2]	A[2]	(A+2)	*(A+2)
&A[3]	A[3]	(A+3)	*(A+3)

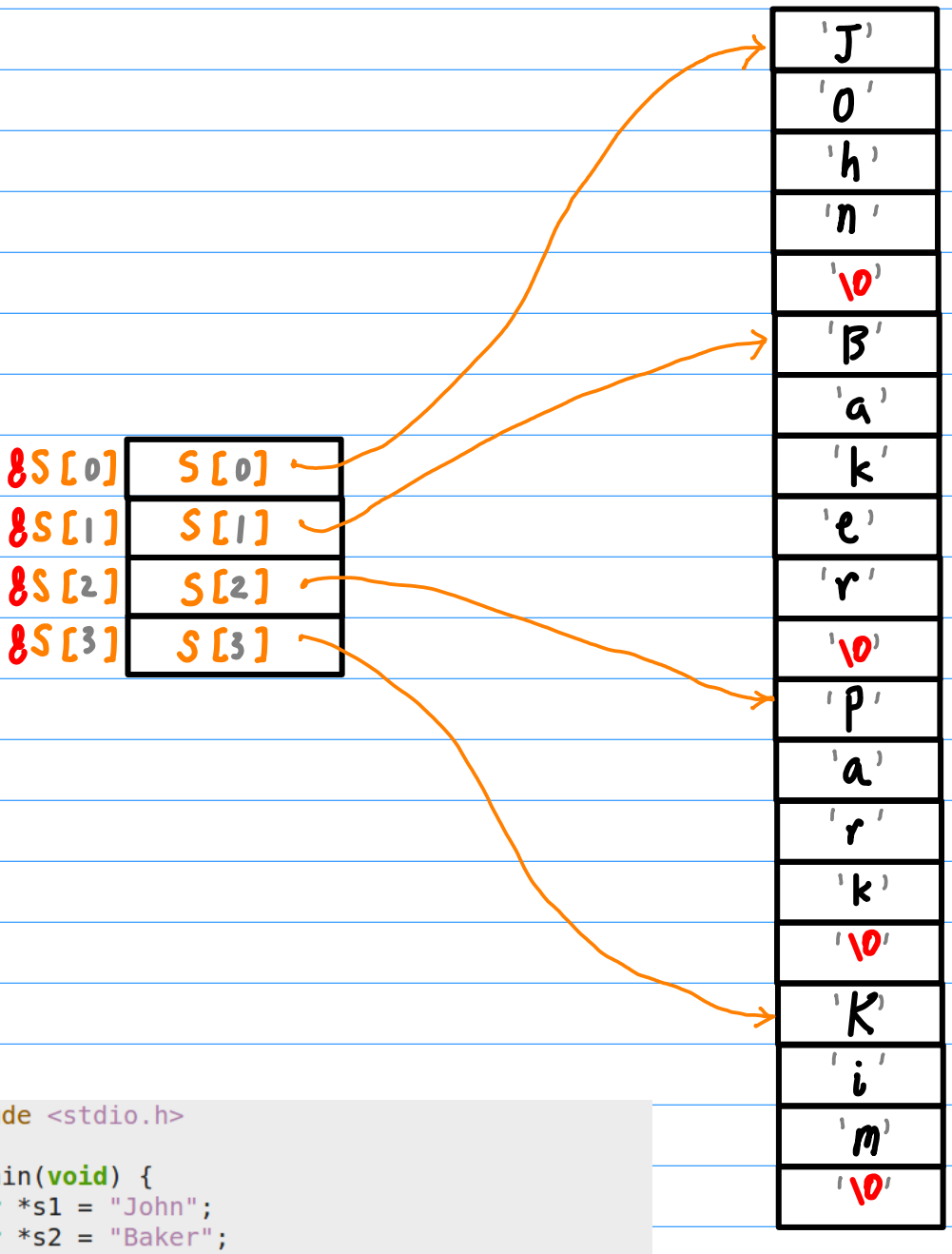
```c
int main(void) {
  int a     =   100;
  int A[4]  = { 100, 200, 300, 400 };

  int *p    =   &a;
  int *P[4] = { A+0, A+1, A+2, A+3 };

  int i;
```

```c
#include <stdio.h>

int main(void) {
  char *s1 = "John";
  char *s2 = "Baker";
  char *s3 = "Park";
  char *s4 = "Kim";

  char *S[4] = { s1, s2, s3, s4 };
  int i;

  printf("s1= %s \n", s1);
  printf("s2= %s \n", s2);
  printf("s3= %s \n", s3);
  printf("s4= %s \n", s4);

  for (i=0; i<4; ++i) {
    printf("S[%d]= %s \n", i, S[i]);
  }

}
```

```
s1= John
s2= Baker
s3= Park
s4= Kim
S[0]= John
S[1]= Baker
S[2]= Park
S[3]= Kim
```




```c
#include <stdio.h>

int main(void) {
  char *s1 = "John";
  char *s2 = "Baker";
  char *s3 = "Park";
  char *s4 = "Kim";

  char *S[4] = { s1, s2, s3, s4 };
  int i;

  printf("s1= %s \n", s1);
  printf("s2= %s \n", s2);
  printf("s3= %s \n", s3);
  printf("s4= %s \n", s4);

  for (i=0; i<4; ++i) {
    printf("S[%d]= %s \n", i, S[i]);
  }

}
```




```c
#include <stdio.h>

int main(void) {
  char *s1 = "John";
  char *s2 = "Baker";
  char *s3 = "Park";
  char *s4 = "Kim";

  char *S[4] = { s1, s2, s3, s4 };
  int i;

  printf("s1= %s \n", s1);
  printf("s2= %s \n", s2);
  printf("s3= %s \n", s3);
  printf("s4= %s \n", s4);

  for (i=0; i<4; ++i) {
    printf("S[%d]= %s \n", i, S[i]);
  }

}
```

The Main Parameters (1)

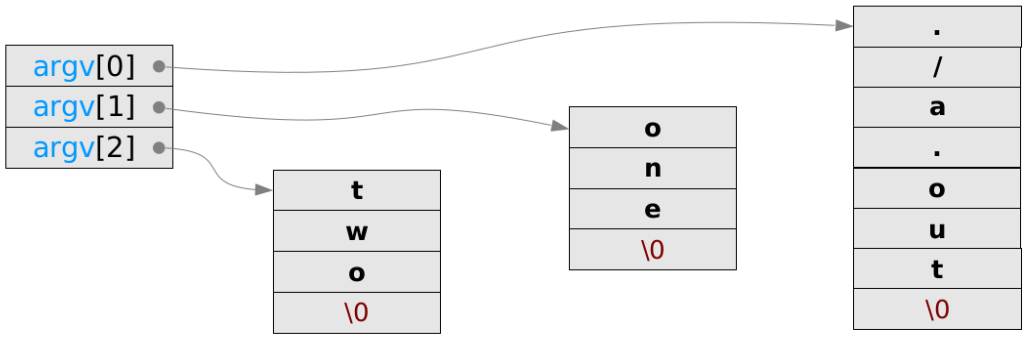
```
#include <stdio.h>

int main(int argc, char *argv[])
{

  int i;

  printf("argc= %d \n", argc);

  for (i=0; i<argc; ++i) {
    printf("argv[%d] = %s \n", i, argv[i]);
  }

}
```

The Main Parameters (2)

```
young@usys ~ $ ./a.out
argc= 1
argv[0] = ./a.out

young@usys ~ $ ./a.out one two three
argc= 4
argv[0] = ./a.out
argv[1] = one
argv[2] = two
argv[3] = three

young@usys ~ $ ./a.out one two three four
argc= 5
argv[0] = ./a.out
argv[1] = one
argv[2] = two
argv[3] = three
argv[4] = four
```

The Main Parameters (3)

```
young@usys ~ $ ./a.out  one  two
argc= 3
argv[0] = ./a.out
argv[1] = one
argv[2] = two
```



```c
#include <stdio.h>

int main( int argc, char * argv[]) {

 printf("Hello\n");

 return 123;

}
```

```
young@USys01 ~ $ ./a.out
Hello
young@USys01 ~ $ echo $?
123
```

main's return value