

Array Pointers (1B)

Copyright (c) 2009-2018 copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

p1.c

```
#include <stdio.h>

int main(void) {
    int a[4] = {1,2,3,4};
    int (*p)[4];

    p = &a;

    printf("( *p)[0]=%d \n", (*p)[0]);
    printf("( *p)[1]=%d \n", (*p)[1]);
    printf("( *p)[2]=%d \n", (*p)[2]);
    printf("( *p)[3]=%d \n", (*p)[3]);

    printf("%ld \n", (a+1) - (a+2));
    printf("%ld \n", (long)(a+1) - (long)(a+2));

    printf("%ld \n", (p+1) - (p+2));
    printf("%ld \n", (long)(p+1) - (long)(p+2));

}

(*p)[0]=1
(*p)[1]=2
(*p)[2]=3
(*p)[3]=4
-1
-4
-1
-16
```

p2.c

```
#include <stdio.h>

void func(int d[])
{
    printf("sizeof(d)=%ld \n", sizeof(d));
    printf("sizeof(*d)=%ld \n", sizeof(*d));
}

int main(void) {
    int a[4];
    int *b;
    int **c;

    int (*p)[4];

    printf("sizeof(a)=%ld \n", sizeof(a));
    printf("sizeof(*a)=%ld \n", sizeof(*a));

    printf("sizeof(b)=%ld \n", sizeof(b));
    printf("sizeof(*b)=%ld \n", sizeof(*b));

    printf("sizeof(c)=%ld \n", sizeof(c));
    printf("sizeof(*c)=%ld \n", sizeof(*c));

    func(a);

    printf("sizeof(p)=%ld \n", sizeof(p));
    printf("sizeof(*p)=%ld \n", sizeof(*p));
}
```

```
sizeof(a)=16
sizeof(*a)=4
sizeof(b)=8
sizeof(*b)=4
sizeof(c)=8
sizeof(*c)=8
sizeof(d)=8
sizeof(*d)=4
sizeof(p)=8
sizeof(*p)=16
```