

Make Example



Systems Programming

A make Example

- Sample Makefile
- magic.h and sub.c
- subB.c
- Arrays.c
- make and arrays.c execution

Sample Makefile

```
OBJECTS= arrays.o sub.o subB.o
arrays: $(OBJECTS)
    gcc $(OBJECTS) -o $@
arrays.o: magic.h
sub.o:
subB.o: magic.h

clean:
    rm -f *.o
```

magic.h and sub.c

magic.h

```
extern int magic;  
#define SIZE 10
```

sub.c

```
#include <stdio.h>  
void sub (int* aptr, int len)  
{  
    int i;  
    printf("S :");  
    for (i =0; i < len; i++)  
    {  
        aptr[i] = 2* (i + 1);  
        printf (" %d", aptr[i]);  
    }  
    printf("\n");  
}
```

subB.c

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

int magic = 22;

void subB (int x, int y, int z, int* aptr, int len)
{
    int i;
    int temp;

    printf ("B :");
    printf (" x = %d , y = %d , z = %d \n", x,y,z);
    temp =z;
    z = y;
    y = x;
    x = temp;
    aptr[len - 1] = 77;
    printf ("B :");
    printf (" x = %d , y = %d , z = %d \n", x,y,z);
```

subB.c (cont.)

```
printf ("B :");  
for (i =0; i < len; i++)  
{  
    printf (" %d", aptr[i]);  
}  
aptr[magic + x] = 33;  
printf("\n");  
}
```

arrays.c

```
#include <stdio.h>
#include "magic.h"
int main()
{
    void sub ();
    void subB ();
    int a[SIZE];
    int i;
    printf("M :");
    for (i = 0; i < SIZE; i++)
        {
            a[i] = i + 1;
            printf(" %d", a[i]);
        }
    printf("\n");
    sub(a, SIZE);
    printf("M2:");
    for (i = 0; i < SIZE/2; i++)
        {
            a[2*i] = 99 - 30*i;
            printf(" %d", a[2*i]);
            printf(" %d", a[2*i + 1]);
        }
    printf("\n");
}
```

arrays.c (cont)

```
sub(a, SIZE);
printf("M2:");
for (i = 0; i < SIZE/2; i++)
{
    a[2*i] = 99 - 30*i;
    printf(" %d", a[2*i]);
    printf(" %d", a[2*i + 1]);
}
printf("\n");
subB(a[6],a[7],a[8], a, SIZE);
printf("M3:");
magic = magic/4;
a[magic] = 45;
for (i = 0; i < SIZE; i++)
{
    printf(" %d", a[i]);
}
printf("\n");
return 0;
}
```


make and arrays execution

```
$ make
```

```
cc -c -o arrays.o arrays.c
```

```
cc -c -o sub.o sub.c
```

```
cc -c -o subB.o subB.c
```

```
gcc arrays.o sub.o subB.o -o arrays
```

```
$ ./arrays
```

```
M: 1 2 3 4 5 6 7 8 9 10
```

```
S: 2 4 6 8 10 12 14 16 18 20
```

```
M2: 99 4 69 8 39 12 9 16 -21 20
```

```
B: x = 9 , y = 16 , z = -21
```

```
B: x = -21 , y = 9 , z = 16
```

```
B: 99 4 69 8 39 12 9 16 -21 77
```

```
M3: 99 33 69 8 39 45 9 16 -21 77
```