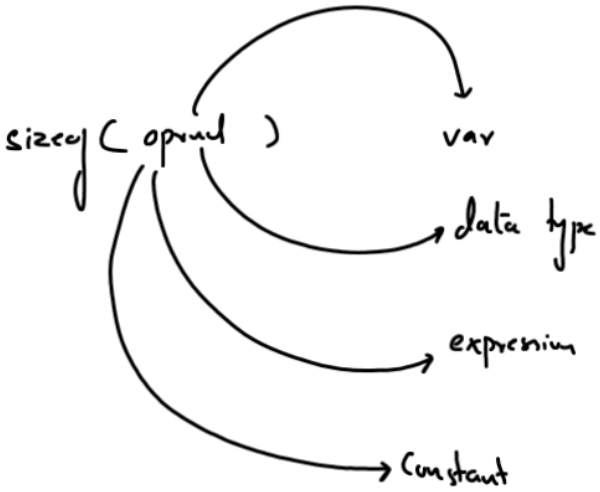
Using "sizeof()"

Syntax:




```
#include <stdio.h>
int main()
{
    int a;
    char b;
    float c;
    printf("size of int is %u bytes",sizeof(a));
    printf("\nsize of char is %u bytes",sizeof(b));
    printf("\nsize of float is %u bytes",sizeof(c));
    return 0;
}
```

OUTPUT

```
=======
size of int is 4 bytes
size of char is 1 bytes
size of float is 4 bytes
```

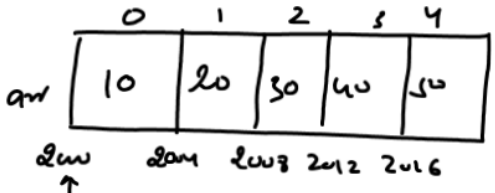
```c
#include <stdio.h>
int main()
{
    printf("size of int is %u bytes",sizeof(int));
    printf("\nsize of char is %u bytes",sizeof(char));
    printf("\nsize of float is %u bytes",sizeof(float));
    return 0;
}
```

OUTPUT

```
=======
size of int is 4 bytes
size of char is 1 bytes
size of float is 4 bytes
```

Accessing Array Using Pointer

```c
int main()
{
    int aw[5];
    int *p;
    int i;
    p=aw;
    for(i=0;i<5;i++)
    {
        printf("Enter no:");
        scanf("%d",p+i);
    }
```



i=0 => 2000+0=>2000

i=1 => 2000+1=>2004

i=2 =>2000+2=>2008

```
for(i=0; i<5; i++)
{
    printf("\n %d", *(p+i));
}

return 0;
}
```

p+i ⟶ p + i * sizeof(data type of ptr)

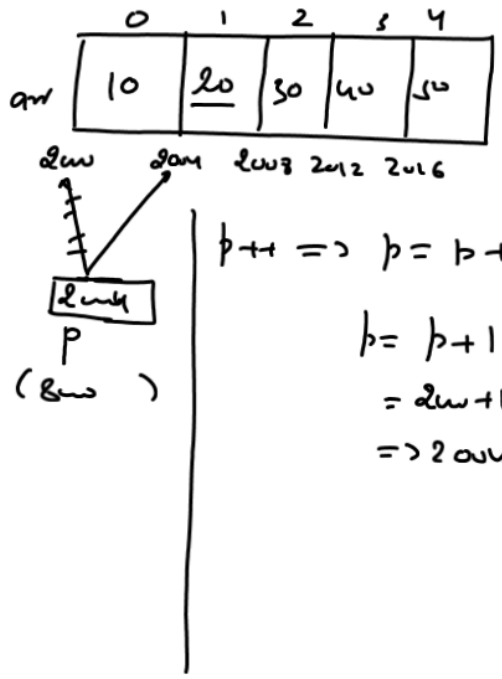
a) i=0 ⟶ 2000 + 0 * 4 => 2000

b) i=1 ⟶ 2000 + 1 * 4 => 2004

c) i=2 ⟶ 2000 + 2 * 4 => 2008

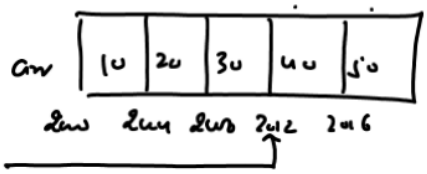
Soln 2

```
int main()
{
    int arr[5];
    int *p;
    int i;
    p = arr;
    for(i=0; i<5; i++)
    {
        printf("Enter no:");
        scanf("%d", p);
        p++;
    }
    p = arr;
    for(i=0; i<5; i++)
    {
        printf("\n%d", *p);
        p++;
    }
return 0;
}
```



How Compiler Actually Handles Array Expression

```
int arr[5] = {10, 20, 30, 40, 50};
```



offset

```
printf("%d", arr[3]);
```
└> 40

```
printf("%d", arr[-1]);
```
? (garbage value at 1996)

arr[i] => *(arr+i)

arr[3] => *(arr+3)
=> *(2000+3)
=> *(2012)
=> 40

*(arr+(-1))
*(2000+(-1*4))
*(1996)