

Discussion 9

Malloc

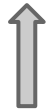
- allocates a block of n bytes of memory (n given as argument)
 - 1 byte is 8 bits
- `void * malloc (size_t size);`



pointer to
memory block



unsigned int



size of block in
bytes (aka. n)

Malloc

- in `stdlib.h`
- Use:
 - `type*` pointer;
 - `type` pointer = `(type cast) malloc (size)`
 - malloc should always be followed by:
 - `if (pointer == NULL)`

ERROR!

- every malloc should pair with:
- `free (pointer);`

ERoberts: in genlib.h

- `pointer = GetBlock(size);`
 - size in bytes

Related: calloc

allocates memory and initializes all elements to zero - very useful for arrays

Example: generate a random character string for a password

```
int i, n;  
char* buffer;  
printf ("Enter size of string");  
scanf ("%d", &i); (or i = GetInteger();)  
buffer = (char *) malloc (sizeof(char) * (i+1));  
if (buffer == NULL) exit (1);  
for (n = 0; n<i;n++)  
    buffer [n] = rand() % 26 + 'a';  
buffer [i] = '\0';  
printf ("%s\n", buffer);  
free (buffer);
```

Structs

```
typedef struct {  
    int x;  
    int y;  
} point;  
point p = {1,3};  
point q;  
q.x = 3;  
q.y = 4;
```