CSE 333 – SECTION 1

C Pointers, Arrays and more

Your TAs

- Chuong Dao
- Undergrad, CSE

- Soumya Vasisht
- Grad, AA

Sections format

- Presentations
- Interactive sessions
- Exercises and worksheets
- Q&A

Let’s C

- General purpose programming language
- Procedural
- Often used in low-level system programming
- Supports use of pointers
- Provides facilities for managing memory
- C passes all of its arguments by value
- Pass-by-reference is simulated by passing the address of a variable

Example

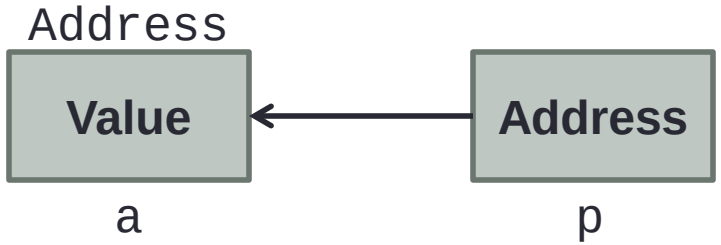
```c
#include <stdio.h>
void f(int *j) {
  (*j)++;
}
int main() {
  int i = 20;
  int *p = &i;
  f(p);
  printf("i = %d\n", i);
  return 0;
}
```

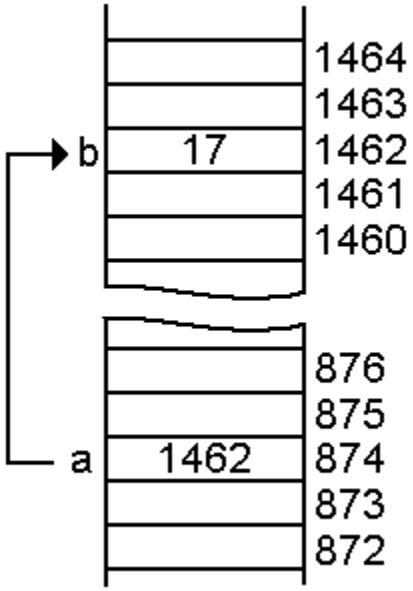
- Output

```
$ gcc test.c
$ ./a.out
i = 21
```

Pointers and Addresses

- A data type that refers to a value stored in another location
- Can point to values, variables of different data types or functions

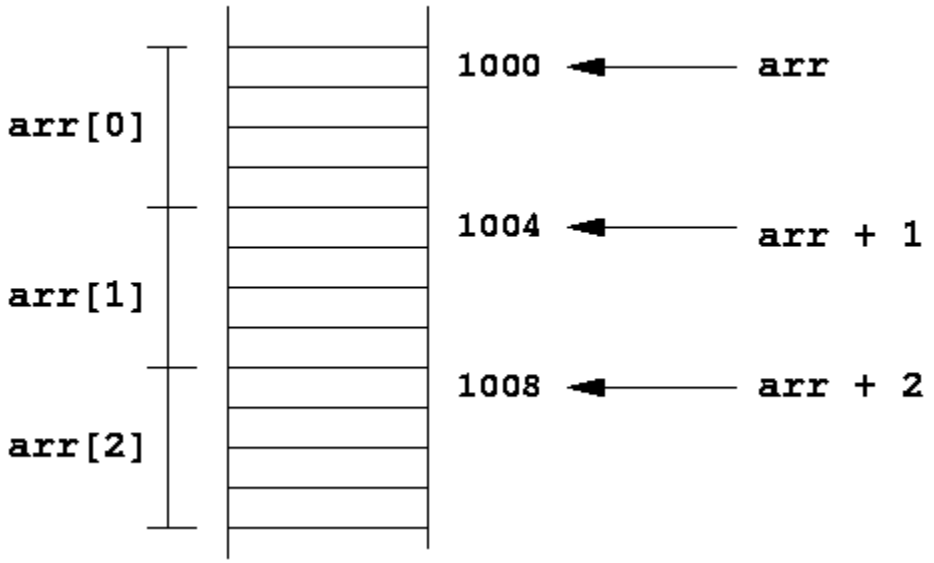




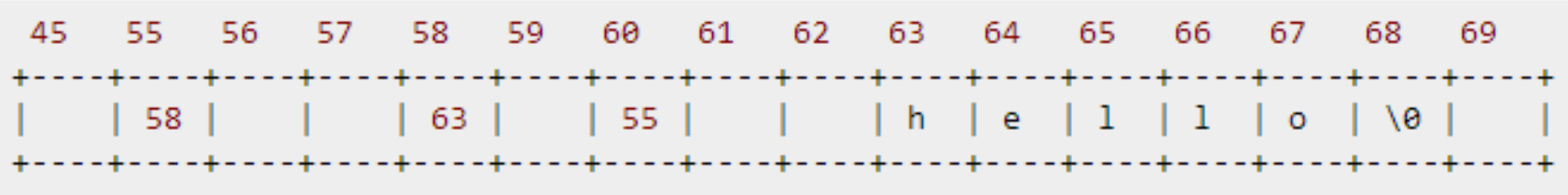
Arrays and pointers

- a[0] <==> *a
- a[3] <==> *(a + 3)

- How about a, a+3,
- *a+3 or *a++?



Pointers to pointers




```
const char *c = “hello”;
const char **cp = &c;
const char ***cpp = &cp;
```

Function pointers

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

 // Function taking a function pointer as an argument
double compute_sum(double (*funcp)(double), double lo, double hi) {
   double sum = 0.0;
  // Add values returned by the pointed-to function '*funcp'
  for (int i = 0; i <= 100; i++) {
    double x, y;
    // Use the function pointer 'funcp' to invoke the function
    x = i/100.0 * (hi - lo) + lo;
    y = (*funcp)(x);
    sum += y;
   }
  return sum / 100;
}
```

```c
int main(void) {
  double (*fp)(double); // Function pointer
  double sum;
  // Use 'sin()' as the pointed-to function
  fp = sin;
  sum = compute_sum(fp, 0.0, 1.0);
  printf("sum(sin): %f\n", sum);
  // Use 'cos()' as the pointed-to function
  fp = cos;
  sum = compute_sum(fp, 0.0, 1.0);
  printf("sum(cos): %f\n", sum);
  return 0;
}
```

Section exercise

- Write a C program that:
- Accepts a string.
- Reverses the string
- Determines whether the string is a palindrome.

- A palindrome is a string which when reversed is same as the original string.
- Ex: abba, aba, mom, noon etc.

Debugging with gdb

- <demo factorial, show gdb features and C style script demo >

Valgrind

- <Valgrind slides>