```c
#include <string.h>
#include <stdio.h>

// Takes two strings a and b, and changes result to have the concatenation of a
// and b stored in it

// ASSUMES that result has length strlen(a) + strlen(b) + 1
void concat(char a[], char b[], char result[]) {
  int alen = strlen(a), blen = strlen(b);
  for(int i = 0; i < alen; i += 1) {
    result[i] = a[i];
  }
  for(int i = 0; i < blen; i += 1) {

    result[alen + i] = b[i];


  }

  result[alen +blen] = '\0';

}

int main() {
  char str1[] = "Hello ";
  char str2[] = "CSE29";

  char result[                ];

  concat(str1, str2, result);

  printf("%s\n", result);


}
```

strlen("Hello ") is 6
str1 array is 7 bytes long

strlen(str1) + strlen(str2) + 1 *

→ what should print?
Hello CSE29

* could use sizeof as well

```
[jpolitz@ieng6-202]:10-10-w2f-concat:501$ gcc concat.c -o concat
concat.c:8:5: error: expected identifier or '(' before '[' token
    8 | char[] concat(char a[], char b[]) {
      |     ^
concat.c: In function 'main':
concat.c:25:18: warning: implicit declaration of function 'concat' [-Wimplicit-function-declar
]
   25 |   printf("%s\n", concat(str1, str2));
      |                  ^~~~~~
concat.c:25:12: warning: format '%s' expects argument of type 'char *', but argument 2 has typ
t' [-Wformat=]
   25 |   printf("%s\n", concat(str1, str2));
      |           ~^     ~~~~~~~~~~~~~~~~~~
      |            |     |
      |            |     int
      |            char *
      |           %d
[jpolitz@ieng6-202]:10-10-w2f-concat:502$
<202.ucsd.edu:/home/linux/ieng6/CSE29_FA25_A00/public/lecture/10-10-w2f-concat]  1,1
```

```c
// ASSUMES that result has length strlen(a) + strlen(b) + 1
char[] concat(char a[], char b[]) {
  char result[strlen(a) + strlen(b) + 1];
  int alen = strlen(a), blen = strlen(b);
  for(int i = 0; i < alen; i += 1) {
    result[i] = a[i];
  }
  for(int i = 0; i < blen; i += 1) {
    result[alen + i] = b[i];
  }
  result[alen + blen] = 0;
  return result;
}
```

cannot return array-typed value from a function in C .

```
// Takes two strings a and b, and changes result to have the concatenation of a
// and b stored in it

// ASSUMES that result has length strlen(a) + strlen(b) + 1
void concat(char a[], char b[], char result[]) {
  int alen = strlen(a), blen = strlen(b);
  for(int i = 0; i < alen; i += 1) {
    result[i] = a[i];
  }
  for(int i = 0; i < blen; i += 1) {
    result[alen + i] = b[i];
  }
  result[alen + blen] = 0;
}
```

"out parameter"

result argument

strcpy(char dest[], char src[])

strncpy

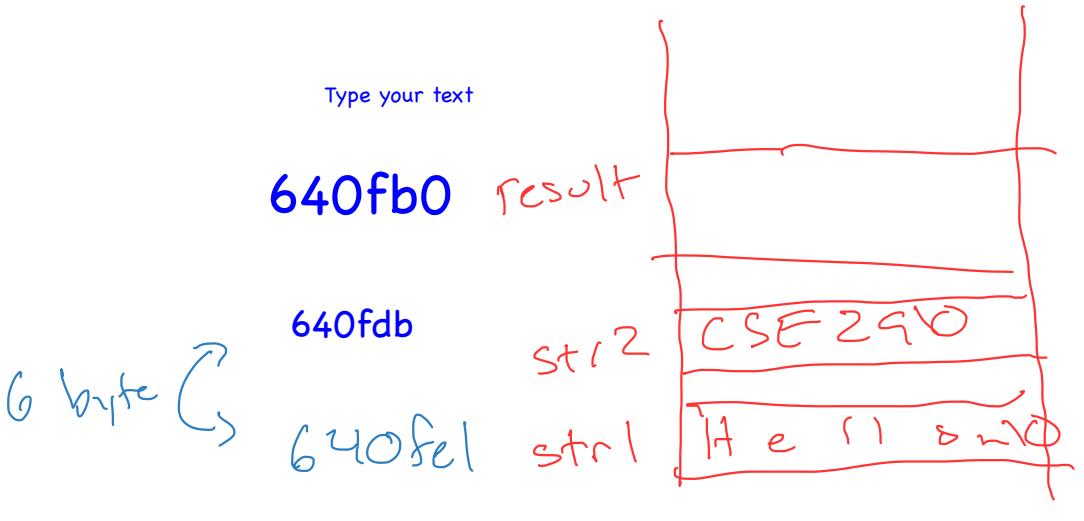
encode_utf8(int cp, char result[])

Addresses in memory where arrays are stored

See them with %p

```
[jpolitz@ieng6-202]:10-10-w2f-concat:504$ gcc concat.c -o concat
[jpolitz@ieng6-202]:10-10-w2f-concat:505$ ./concat
0x7ffeea640fe1 0x7ffeea640fdb 0x7ffeea640fb0
Hello CSE29
0x7ffeea640fe1 0x7ffeea640fdb 0x7ffeea640fb0
[jpolitz@ieng6-202]:10-10-w2f-concat:506$
<202.ucsd.edu:/home/linux/ieng6/CSE29_FA25_A00/public/lecture/10-1
```

```c
// ASSUMES that result has length strlen(a) + strlen(b) + 1
void concat(char a[], char b[], char result[]) {
  printf("%p %p %p\n", a, b, result);
  int alen = strlen(a), blen = strlen(b);
  for(int i = 0; i < alen; i += 1) {
    result[i] = a[i];
  }
  for(int i = 0; i < blen; i += 1) {
    result[alen + i] = b[i];
  }
  result[alen + blen] = 0;
}


int main() {
  char str1[] = "Hello ";
  char str2[] = "CSE29";

  char result[strlen(str1) + strlen(str2) + 1];

  concat(str1, str2, result);

  printf("%s\n", result);

  printf("%p %p %p\n", str1, str2, result);


}
```



db
dc
dd
de
df
e0
e1

b0
bf
} 15 bytes

c0
cf
} 15

d0
d1
d2
d3
...