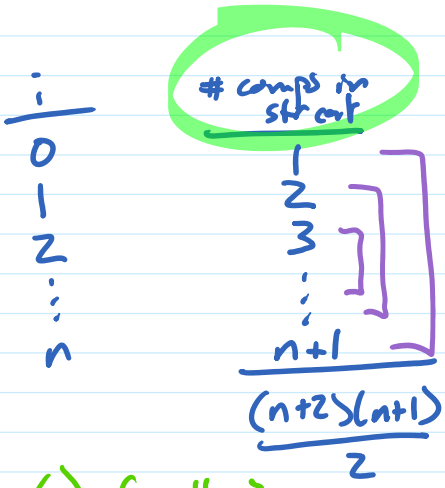
```
make_banner(wall, "#", 1000000);
...
void make_banner(char dest[], char src[], int n)
{
  strcpy(dest, "");
  int len = strlen(src);
  for (int i = 0; i < n; i++)
    {
      strcat(dest, src);
    }
}
```

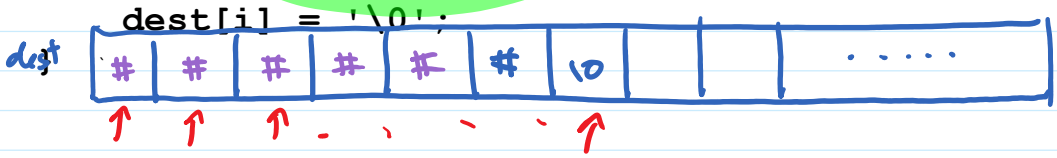


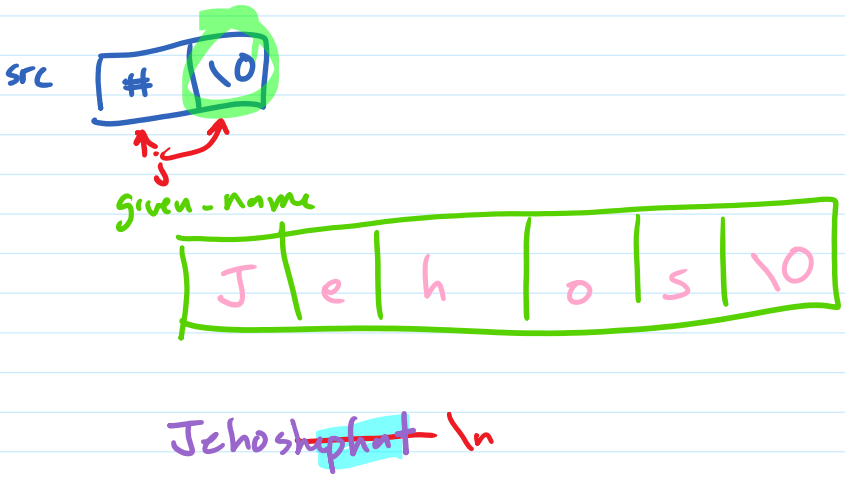
$\frac{n^2}{2} + \frac{3}{2}n + 1$ vs $1000n^2$ vs $\Theta(n^2)$ vs $10^2 \cdot n$ vs $1000000n$ vs $\Theta(n)$

there is a large enough n_0 s.t. $f(n) \geq g(n)$ for all $n > n_0$

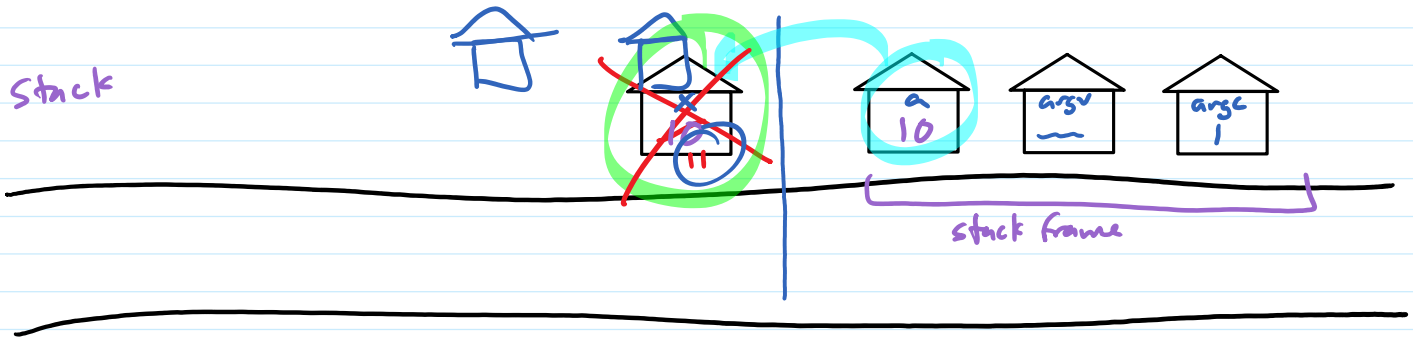
```
void strcat(char dest[], char src[]) {
    int i = 0;
    while (dest[i] != '\0') {
     i++;
    }
    int j = 0;
    while (src[j] != '\0') { dest[i] = src[j]; i++; j++; }
    dest[i] = '\0';
```

for n-char dest, n+1 executions



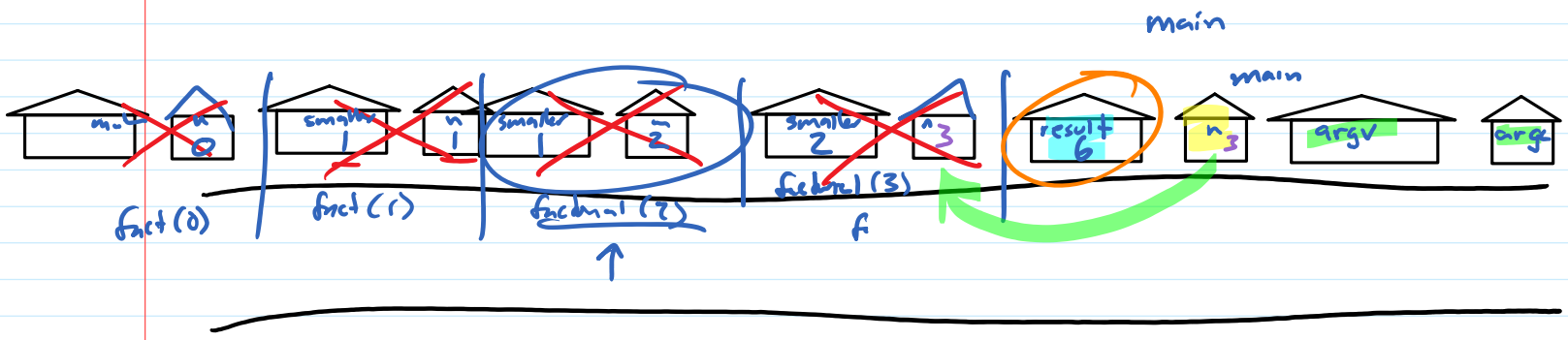


Jehoshaphat \n



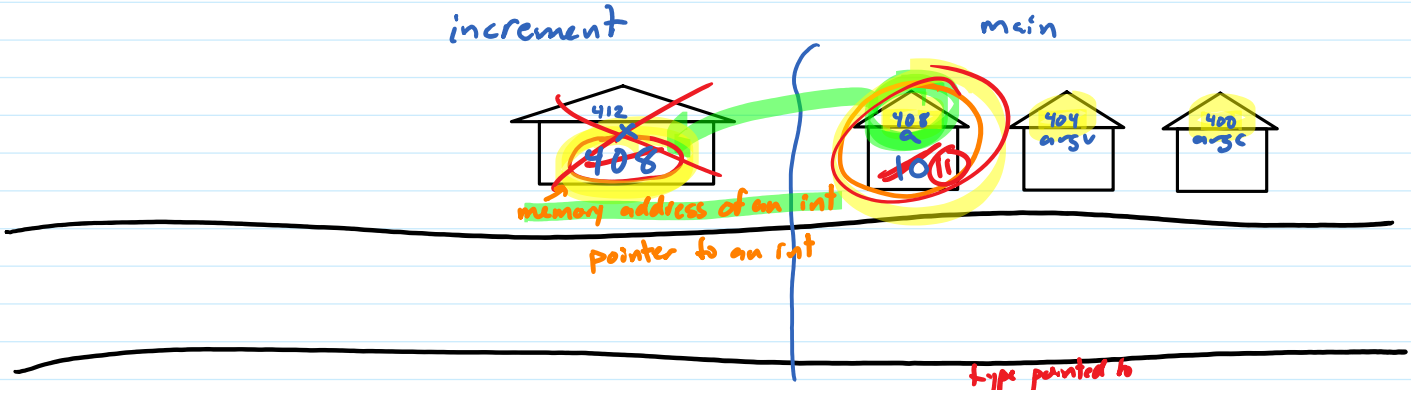

```
int main(int argc, char *argv[]) {
  int a = 10;
  increment(a);
  printf("a=%d\n", a);
}
```

```
int increment(int x) {
  x = x + 1;
}
```




```
int main(int argc, char *argv[]) {
  int n;
  scanf("%d", &n);
  long result = factorial(n);
  printf("%d! = %ld\n", n, result);
}
```

```
long factorial(int n) {
  if (n == 0)
    return 1;
  else {
    long smaller = factorial(n - 1);
    return n * smaller;
  }
}
```



```
int main(int argc, char *argv[]) {
  int a = 10;
  increment(&a);
  printf("a=%d\n", a);
}
```

```
void increment(int *x) {
  *x = *x + 1;
}
```

type pointed to

10 + 1

11