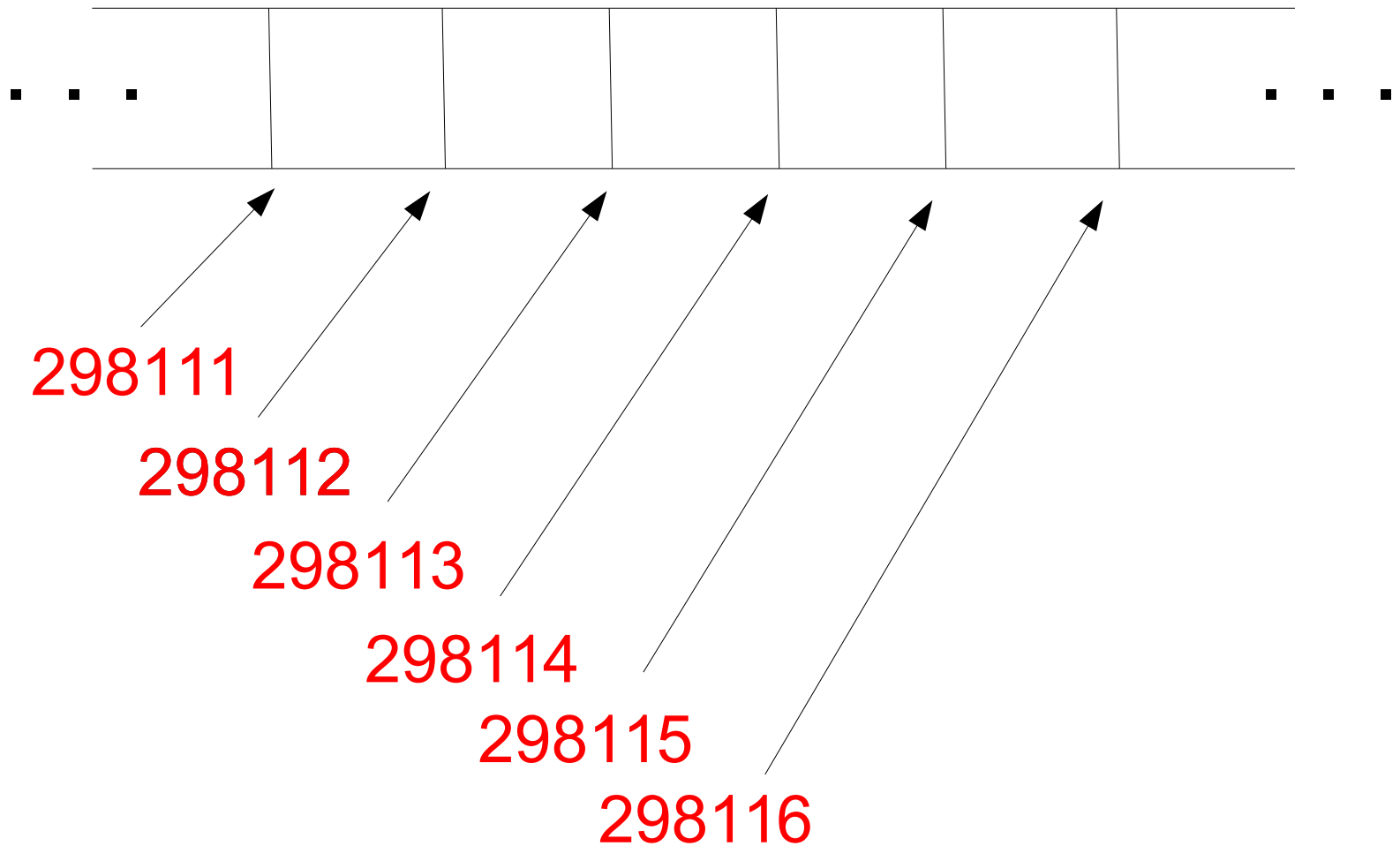
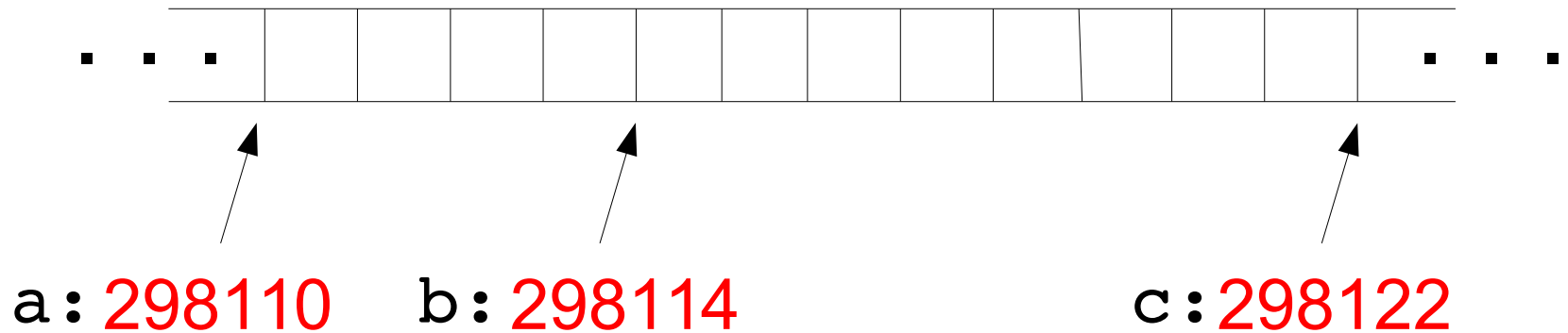


# How Variables Are Stored in Memory



Every byte has an address – numbered consecutively

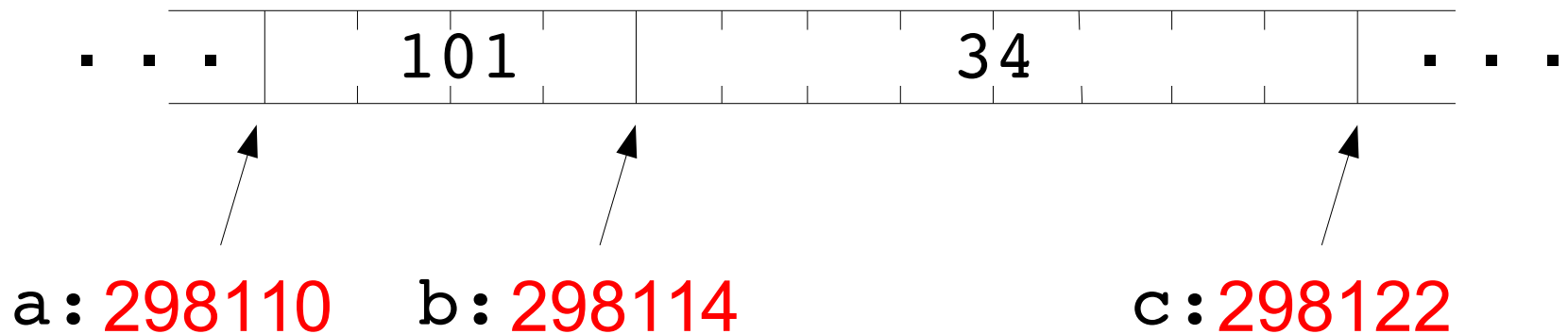
# How Variables Are Stored in Memory



```
int a;    /* 4 bytes for an int */
long b;   /* 8 bytes for a long */
int c;    /* 4 bytes for an int */
```

Space is reserved for each variable

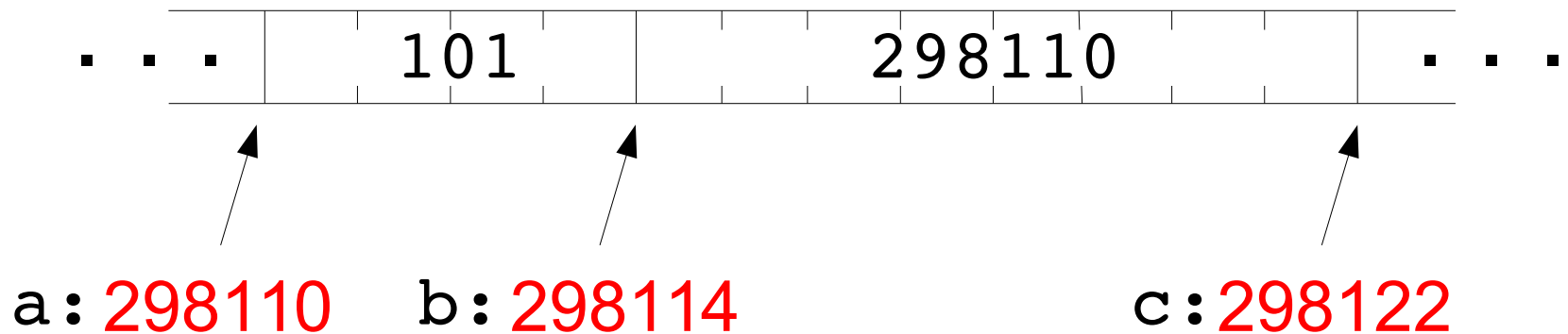
# How Variables Are Stored in Memory



```
int a = 101;    /* 4 bytes for an int */
long b = 34;   /* 8 bytes for a long */
int c;         /* 4 bytes for an int */
```

Variable values are stored in reserved spaces

# How Variables Are Stored in Memory



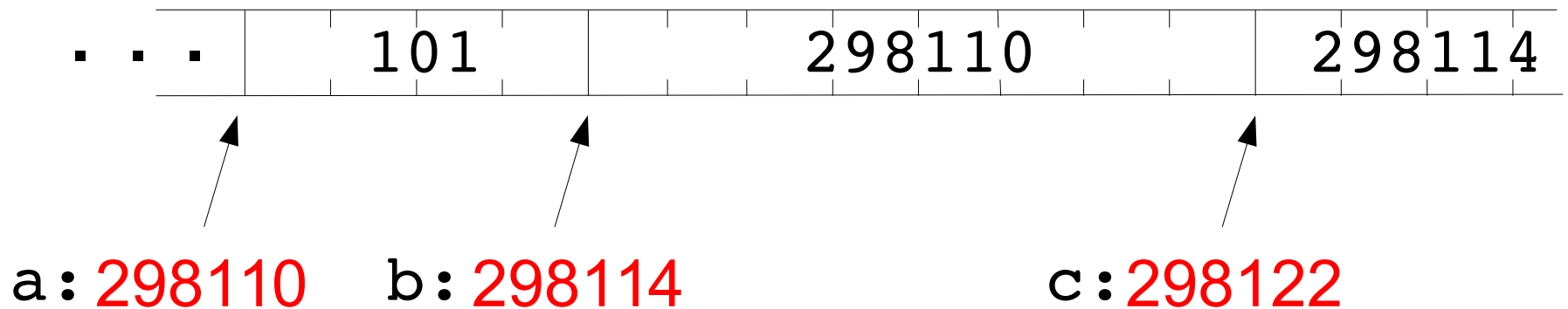
```
int a = 101;      /* 4 bytes for an int */
int *b = &a;     /* 8 bytes for an int* */
int c;           /* 4 bytes for an int */

cout << b << " " << *b << "\n";
298110  101
```

dereference pointer

Variable values are stored in reserved spaces

# How Variables Are Stored in Memory



```
int a = 101; /* 4 bytes for an int */
int *b = &a; /* 8 bytes for an int* */
int **c = &b; /* 8 bytes for an int** */
cout << c <<" " << *c <<" " << **c << "\n";
298114 298110 101
```

Variable values are stored in reserved spaces

# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}
```

```
void g(int x) {  
    cout << x+1 << "\n";  
}
```

```
int main () {  
    f(201);  
    return 1;  
}
```

Stack

---

One stack per thread

# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}
```

```
void g(int x) {  
    cout << x+1 << "\n";  
}
```

```
int main () {  
    f(201);  
    return 1;  
}
```

Stack

Return address in main

One stack per thread

# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}
```

```
void g(int x) {  
    cout << x+1 << "\n";  
}
```

```
int main () {  
    f(201);  
    return 1;  
}
```

Stack

Return address in main

x = 201

One stack per thread



# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}  
  
void g(int x) {  
    cout << x+1 << "\n";  
}  
  
int main () {  
    f(201);  
    return 1;  
}
```

Stack

Return address in main

x = 201

y = 101

One stack per thread

# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}  
  
void g(int x) {  
    cout << x+1 << "\n";  
}  
  
int main () {  
    f(201);  
    return 1;  
}
```

Stack

---

Return address in main

---

x = 201  
y = 101

---

Return address in f

---

One stack per thread

# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}  
  
void g(int x) {  
    cout << x+1 << "\n";  
}  
  
int main () {  
    f(201);  
    return 1;  
}
```

Stack

---

Return address in main

---

x = 201  
y = 101

---

Return address in f

---

x = 20301

One stack per thread

# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}  
  
void g(int x) {  
    cout << x+1 << "\n";  
}  
  
int main () {  
    f(201);  
    return 1;  
}
```

Stack

---

Return address in main

---

x = 201  
y = 101

---

Return address in f

---

x = 20301

One stack per thread

# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}  
  
void g(int x) {  
    cout << x+1 << "\n";  
}  
  
int main () {  
    f(201);  
    return 1;  
}
```

Stack

---

Return address in main

---

x = 201  
y = 101

---

Return address in f

---

One stack per thread

# How Variables Are Stored in Memory

```
void f(int x) {
    int y = 101;
    g(x*y);
    g(x+y);
}

void g(int x) {
    cout << x+1 << "\n";
}

int main () {
    f(201);
    return 1;
}
```

Stack

---

Return address in main

---

x = 201

y = 101

---

One stack per thread

# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}  
  
void g(int x) {  
    cout << x+1 << "\n";  
}  
  
int main () {  
    f(201);  
    return 1;  
}
```

Stack

---

Return address in main

---

x = 201  
y = 101

---

Return address in f

---

One stack per thread

# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}  
  
void g(int x) {  
    cout << x+1 << "\n";  
}  
  
int main () {  
    f(201);  
    return 1;  
}
```

Stack

---

Return address in main

---

x = 201  
y = 101

---

Return address in f

---

x = 302

One stack per thread



# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}  
  
void g(int x) {  
    cout << x+1 << "\n";  
}  
  
int main () {  
    f(201);  
    return 1;  
}
```

Stack

---

Return address in main

---

x = 201  
y = 101

---

Return address in f

---

x = 302

One stack per thread

# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}  
  
void g(int x) {  
    cout << x+1 << "\n";  
}  
  
int main () {  
    f(201);  
    return 1;  
}
```

Stack

---

Return address in main

---

x = 201  
y = 101

---

Return address in f

---

One stack per thread

# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}  
  
void g(int x) {  
    cout << x+1 << "\n";  
}  
  
int main () {  
    f(201);  
    return 1;  
}
```

Stack

---

Return address in main

---

x = 201  
y = 101

---

One stack per thread

# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}
```

```
void g(int x) {  
    cout << x+1 << "\n";  
}
```

```
int main () {  
    f(201);  
    return 1;  
}
```

Stack

Return address in main

One stack per thread

# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}
```

```
void g(int x) {  
    cout << x+1 << "\n";  
}
```

```
int main () {  
    f(201);  
    return 1;  
}
```

Stack

---

One stack per thread

# How Variables Are Stored in Memory

```
void f(int x) {  
    int y = 101;  
    g(x*y);  
    g(x+y);  
}
```

```
void g(int x) {  
    cout << x+1 << "\n";  
}
```

```
int main () {  
    f(201);  
    return 1;  
}
```

Stack

---

One stack per thread

# How Variables Are Stored in Memory

```
void f() {  
    int *y = new int(4);  
    g(y);  
}
```

```
void g(int *x) {  
    cout << *x+1 << "\n";  
}
```

```
int main () {  
    f();  
    return 1;  
}
```

Stack

---

Heap

---

One stack per thread

# How Variables Are Stored in Memory

```
void f() {  
    int *y = new int(4);  
    g(y);  
}
```

```
void g(int *x) {  
    cout << *x+1 << "\n";  
}
```

```
int main () {  
    f();  
    return 1;  
}
```

Stack

Return address in main

Heap

One stack per thread



# How Variables Are Stored in Memory

```
void f() {  
    int *y = new int(4);  
    g(y);  
}  
  
void g(int *x) {  
    cout << *x+1 << "\n";  
}  
  
int main () {  
    f();  
    return 1;  
}
```

Stack

---

Return address in main

---

y = 98221332

---

Heap

---

98221332: 4

One stack per thread

# How Variables Are Stored in Memory

```
void f() {  
    int *y = new int(4);  
    g(y);  
}  
  
void g(int *x) {  
    cout << *x+1 << "\n";  
}  
  
int main () {  
    f();  
    return 1;  
}
```

Stack

---

Return address in main

.....

y = 98221332

---

Return address in f

.....

Heap

---

98221332: 4

One stack per thread

# How Variables Are Stored in Memory

```
void f() {
    int *y = new int(4);
    g(y);
}

void g(int *x) {
    cout << *x+1 << "\n";
}

int main () {
    f();
    return 1;
}
```

Stack

Return address in main

y = 98221332

Return address in f

x = 98221332

Heap

98221332: 4

One stack per thread

# How Variables Are Stored in Memory

```
void f() {  
    int *y = new int(4);  
    g(y);  
}  
  
void g(int *x) {  
    cout << *x+1 << "\n";  
}  
  
int main () {  
    f();  
    return 1;  
}
```

Stack

Return address in main

y = 98221332

Return address in f

x = 98221332

Heap

98221332: 4

One stack per thread

# How Variables Are Stored in Memory

```
void f() {
    int *y = new int(4);
    g(y);
}

void g(int *x) {
    cout << *x+1 << "\n";
}

int main () {
    f();
    return 1;
}
```

Stack

Return address in main

y = 98221332

Return address in f

Heap

98221332: 4

One stack per thread

# How Variables Are Stored in Memory

```
void f() {  
    int *y = new int(4);  
    g(y);  
}  
  
void g(int *x) {  
    cout << *x+1 << "\n";  
}  
  
int main () {  
    f();  
    return 1;  
}
```

Stack

---

Return address in main

---

y = 98221332

---

Heap

---

98221332: 4

One stack per thread

# How Variables Are Stored in Memory

```
void f() {  
    int *y = new int(4);  
    g(y);  
}
```

```
void g(int *x) {  
    cout << *x+1 << "\n";  
}
```

```
int main () {  
    f();  
    return 1;  
}
```

Stack

Return address in main

Heap

98221332: 4

One stack per thread

# How Variables Are Stored in Memory

```
void f() {  
    int *y = new int(4);  
    g(y);  
}  
  
void g(int *x) {  
    cout << *x+1 << "\n";  
}  
  
int main () {  
    f();  
    return 1;  
}
```

Stack

---

Heap

---

98221332: 4

One stack per thread



# How Variables Are Stored in Memory

```
void f() {  
    int *y = new int(4);  
    g(y);  
}  
  
void g(int *x) {  
    cout << *x+1 << "\n";  
}  
  
int main () {  
    f();  
    return 1;  
}
```

Stack

---

Heap

---

98221332: 4

One stack per thread