

- [CPSC 275: Introduction to Computer Systems](#)

## [CPSC 275: Introduction to Computer Systems](#)

Fall 2025

- [Syllabus](#)
- [Schedule](#)
- [Resources](#)
- [Upload](#)
- [Solution](#)

# Solution to Homework 21

1. We first annotate the assembly code:

```
movl 8(%ebp),%ecx      # Get i
movl 12(%ebp),%edx     # Get j
leal 0(,%ecx,8),%eax   # 8*i
subl %ecx,%eax        # 8*i-i = 7*i
addl %edx,%eax        # 7*i+j
leal (%edx,%edx,4),%edx # 5*j
addl %ecx,%edx        # 5*j+i
movl mat1(,%eax,4),%eax # mat1[7*i+j]
addl mat2(,%edx,4),%eax # mat2[5*j+i]
```

We can see that the reference to matrix mat1 is at byte offset  $4(7i + j)$ , while the reference to matrix mat2 is at byte offset  $4(5j + i)$ . From this, we can determine that mat1 has 7 columns, while mat2 has 5, giving  $M = 5$  and  $N = 7$ .

2. Array element  $A[i][j][k]$  is located at address  $x_A + 4(T(S \cdot i + j) + k)$ , where  $x_A$  is the beginning address of A.

- 3.
- A. customerRecord.lastName
  - B. customerPtr->lastName
  - C. customerRecord.firstName
  - D. customerPtr->firstName
  - E. customerRecord.customerNumber
  - F. customerPtr->customerNumber
  - G. customerRecord.personal.phoneNumber
  - H. customerPtr->personal.phoneNumber
  - I. customerRecord.personal.address
  - J. customerPtr->personal.address
  - K. customerRecord.personal.city
  - L. customerPtr->personal.city
  - M. customerRecord.personal.state
  - N. customerPtr->personal.state
  - O. customerRecord.personal.zipCode
  - P. customerPtr->personal.zipCode

4.

A. The layout of the structure is as follows:

Offset	0	4	8	12	16
Contents	p	s.x	s.y	next	

B. It uses 16 bytes.

C. We start by annotating the assembly code:

```

                                # sp at %ebp+8
movl 8(%ebp), %eax             # Get sp
movl 8(%eax), %edx             # Get sp->s.y
movl %edx, 4(%eax)             # Store in sp->s.x
leal 4(%eax), %edx             # Compute &(sp->s.x)
movl %edx, (%eax)              # Store in sp->p
movl %eax, 12(%eax)            # Store sp in sp->next

```

From this, we can generate C code as follows:

```

void sp_init(struct prob *sp)
{
    sp->s.x = sp->s.y;
    sp->p = &(sp->s.x);
    sp->next = sp;
}

```

5.

- **Welcome: Sean**

- [LogOut](#)

