Computer Programming

Dr. Deepak B Phatak
Dr. Supratik Chakraborty
Department of Computer Science and Engineering
IIT Bombay

Session: Use of Pointers In C++ Programs

Quick Recap of Relevant Topics

- Basic programming constructs
- Pointer data type in C++
- “Address of” operator in C++
- “Content of” operator in C++
- Caveats when using “address of” and “contents of”

Overview of This Lecture

- Understanding usage of pointers in C++ programs
- Understanding how contents of memory locations change when a program with pointers executes

A C++ Program With Pointers

IIT Bombay

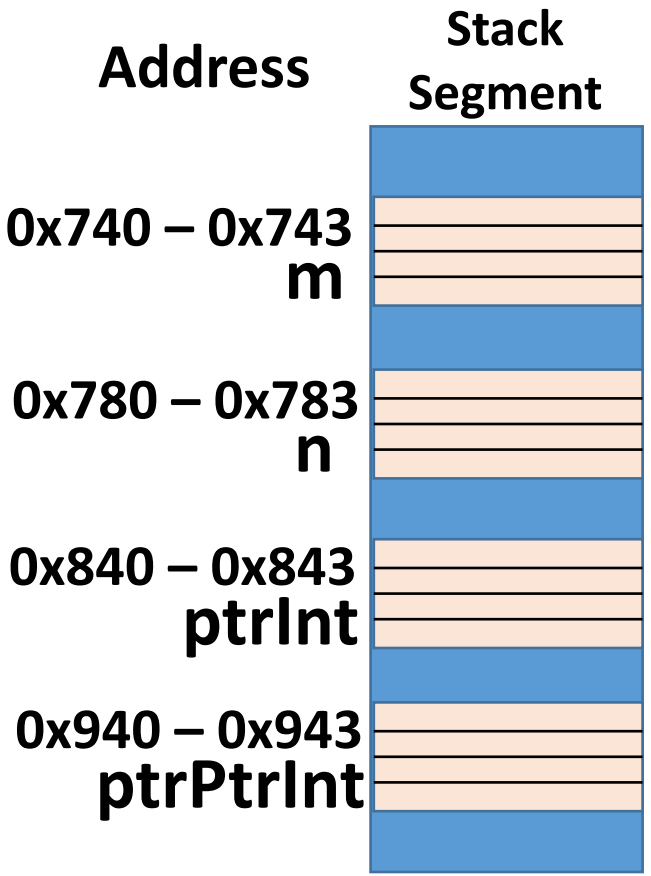
```
int main()
{  int m, n;
   int * ptrInt;
   int ** ptrPtrInt;
   ptrPtrInt = &ptrInt;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; cout << *(*ptrPtrInt) << endl;
   ptrInt = &m;  cout << *(*ptrPtrInt) << endl;
   return 0;
}
```

If we give input values of 5 and 6 for m and n, what does the program output?

A C++ Program With Pointers

IIT Bombay

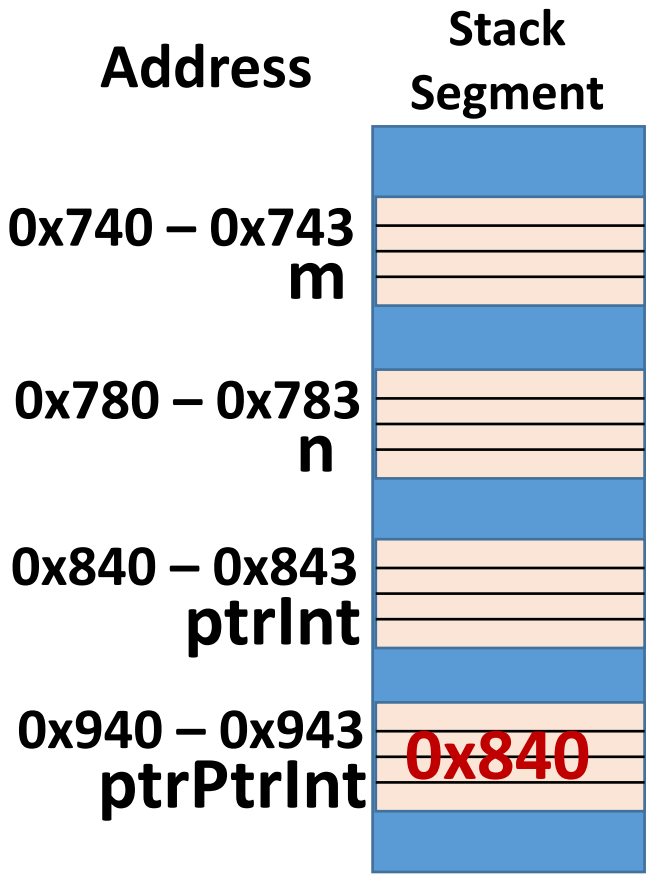
```
int main()
{  int m, n;
   int * ptrInt;
   int ** ptrPtrInt;
   ptrPtrInt = &ptrInt;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; cout << *(*ptrPtrInt) << endl;
   ptrInt = &m;  cout << *(*ptrPtrInt) << endl;
   return 0;
}
```



A C++ Program With Pointers

IIT Bombay

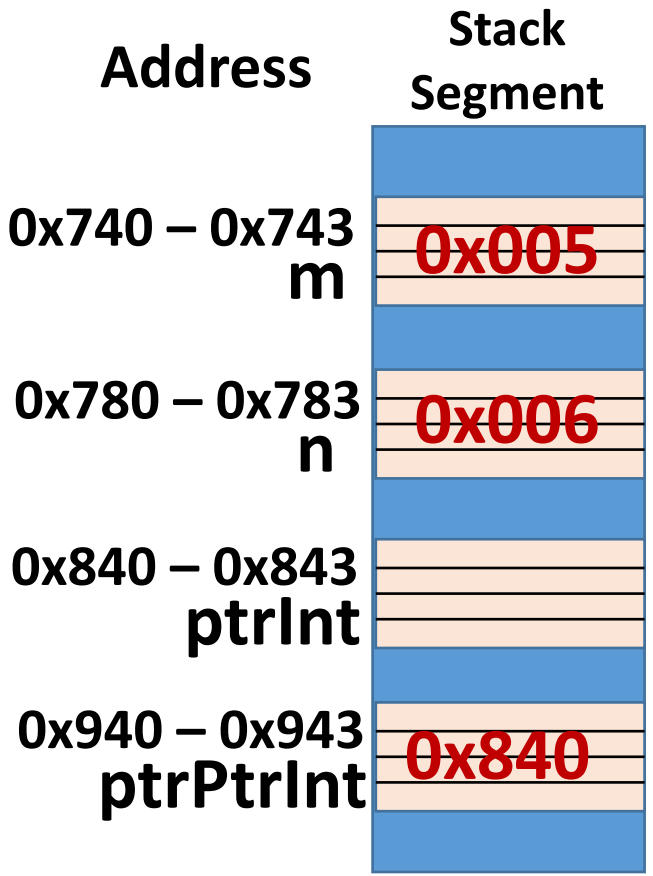
```
int main()
{  int m, n;
   int * ptrInt;
   int ** ptrPtrInt;
   ptrPtrInt = &ptrInt;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; cout << *(*ptrPtrInt) << endl;
   ptrInt = &m;  cout << *(*ptrPtrInt) << endl;
   return 0;
}
```



A C++ Program With Pointers

IIT Bombay

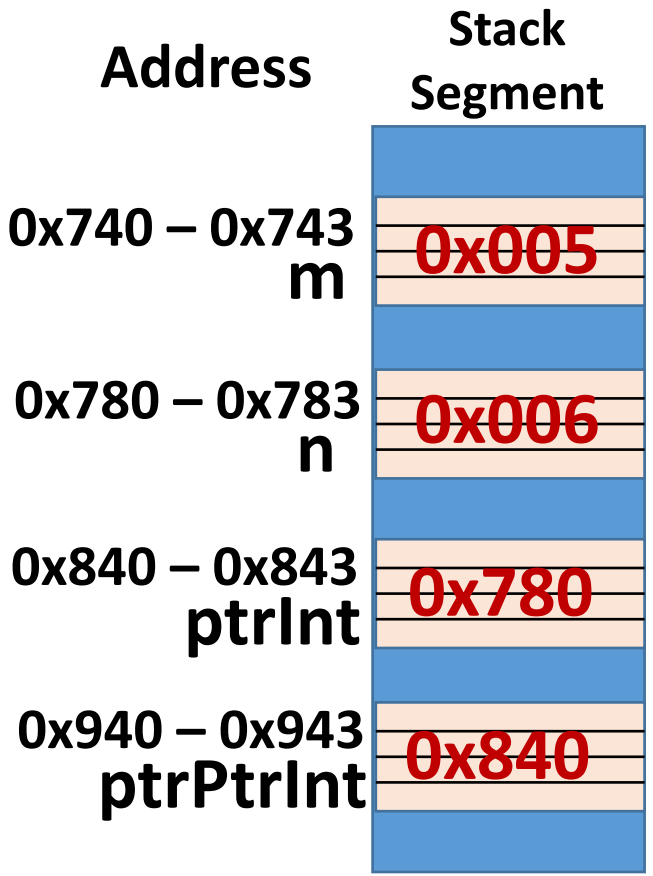
```
int main()
{  int m, n;
   int * ptrInt;
   int ** ptrPtrInt;
   ptrPtrInt = &ptrInt;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; cout << *(*ptrPtrInt) << endl;
   ptrInt = &m;  cout << *(*ptrPtrInt) << endl;
   return 0;
}
```



A C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n;
   int * ptrInt;
   int ** ptrPtrInt;
   ptrPtrInt = &ptrInt;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; cout << *(*ptrPtrInt) << endl;
   ptrInt = &m;  cout << *(*ptrPtrInt) << endl;
   return 0;
}
```

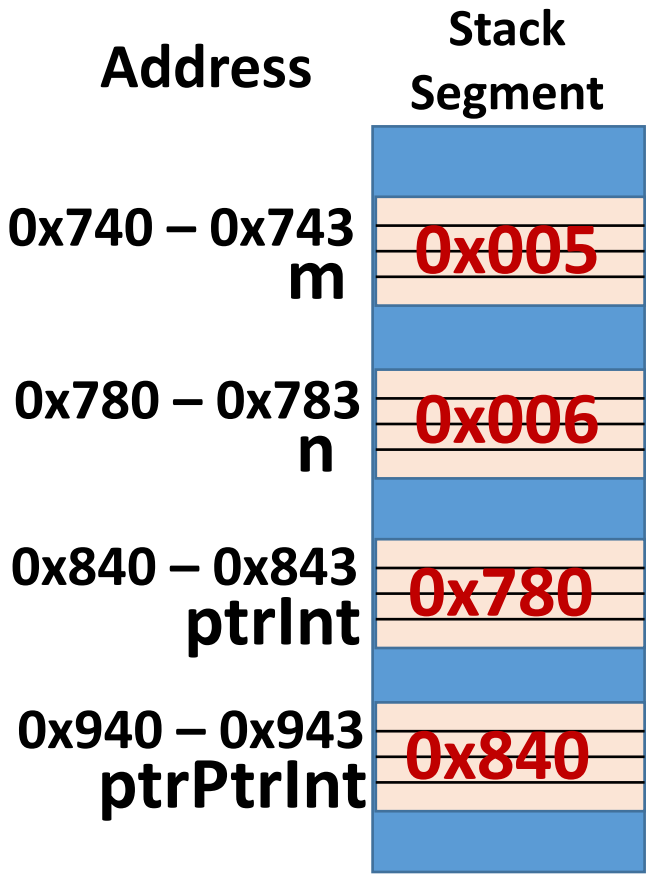


A C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n;
   int * ptrInt;
   int ** ptrPtrInt;
   ptrPtrInt = &ptrInt;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; cout << *(*ptrPtrInt) << endl;
   ptrInt = &m;  cout << *(*ptrPtrInt) << endl;
   return 0;
}
```

Content of (content of ptrPtrInt)

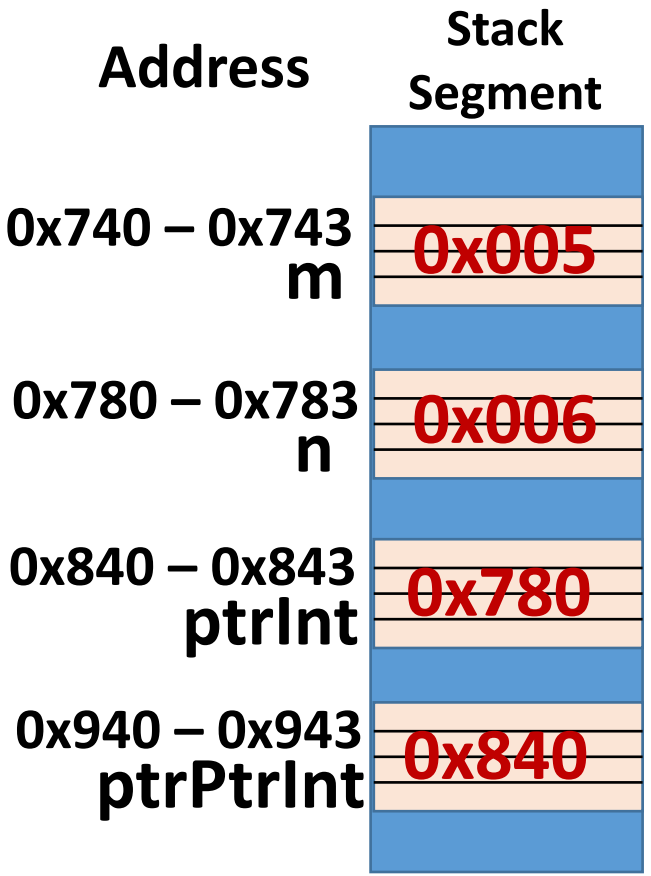


A C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n;
   int * ptrInt;
   int ** ptrPtrInt;
   ptrPtrInt = &ptrInt;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; cout << *(*ptrPtrInt) << endl;
   ptrInt = &m;  cout << *(*ptrPtrInt) << endl;
   return 0;
}
```

Content of (0x780)

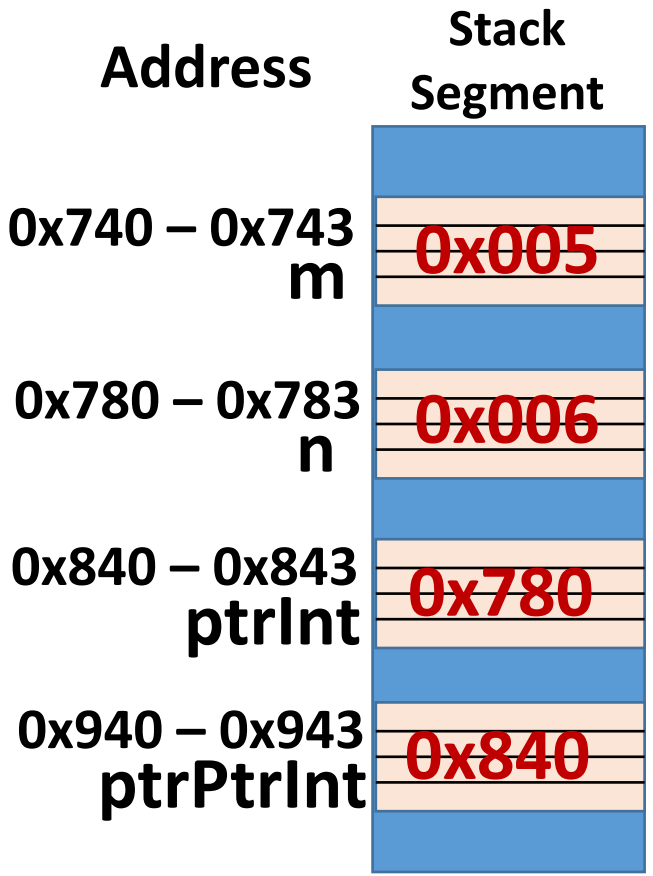


A C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n;
   int * ptrInt;
   int ** ptrPtrInt;
   ptrPtrInt = &ptrInt;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; cout << *(*ptrPtrInt) << endl;
   ptrInt = &m;  cout << *(*ptrPtrInt) << endl;
   return 0;
}
```

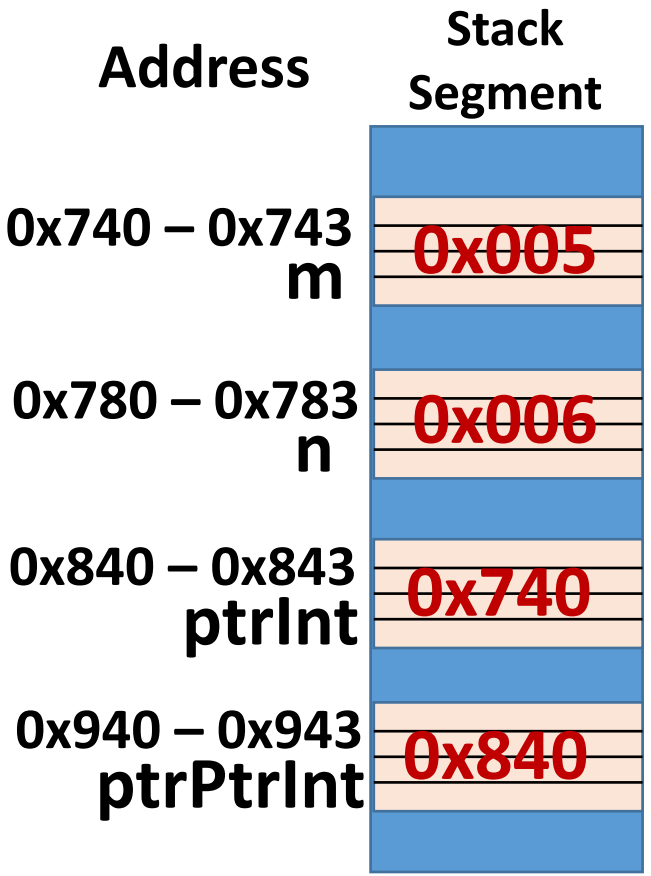
Print 6



A C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n;
   int * ptrInt;
   int ** ptrPtrInt;
   ptrPtrInt = &ptrInt;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; cout << *(*ptrPtrInt) << endl;
   ptrInt = &m;  cout << *(*ptrPtrInt) << endl;
   return 0;
}
```

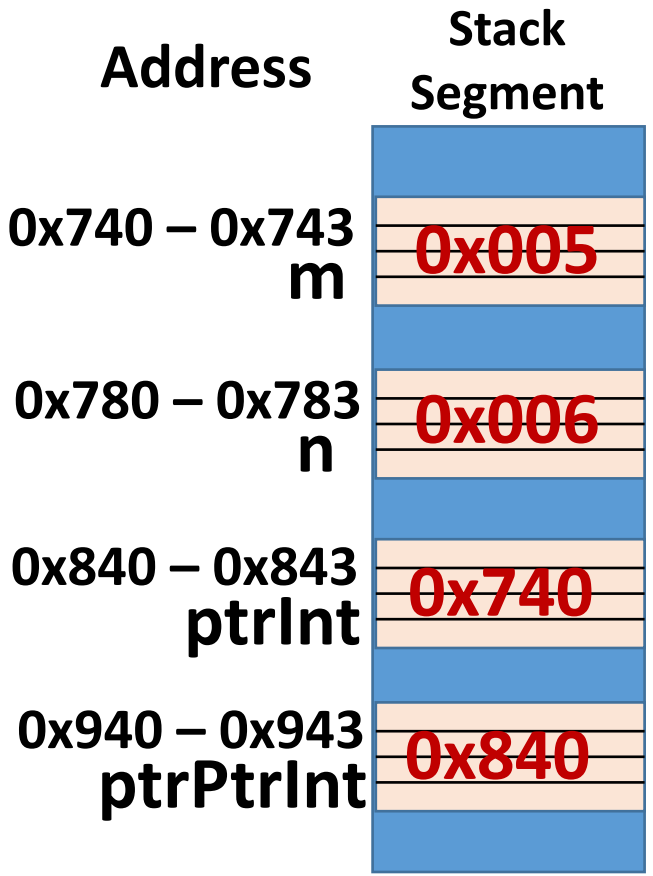


A C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n;
   int * ptrInt;
   int ** ptrPtrInt;
   ptrPtrInt = &ptrInt;
   cout << "Give m and n"; cin >> m >> n;
   ptrInt = &n; cout << *(*ptrPtrInt) << endl;
   ptrInt = &m; cout << *(*ptrPtrInt) << endl;
   return 0;
}
```

Content of (content of ptrPtrInt)

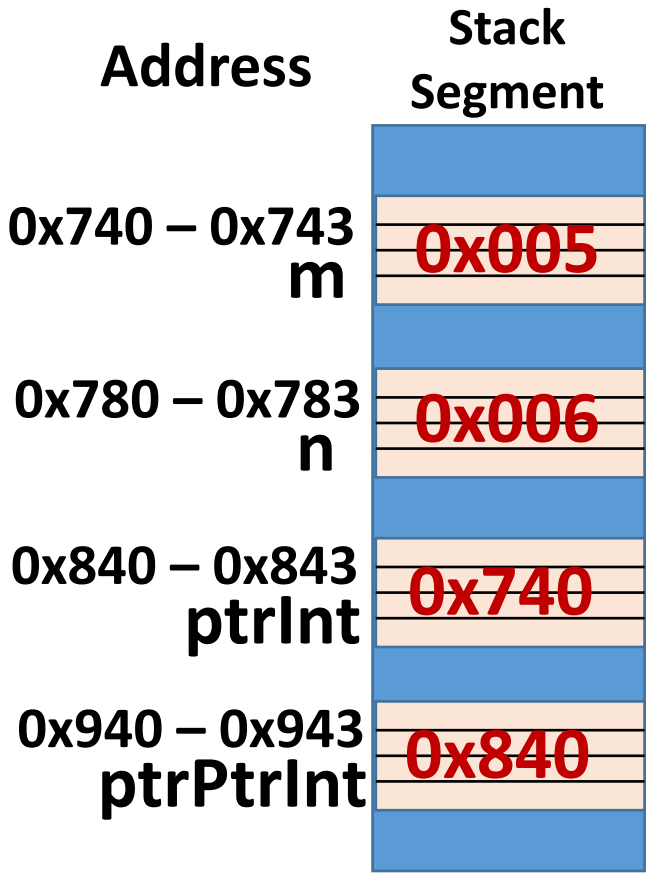


A C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n;
   int * ptrInt;
   int ** ptrPtrInt;
   ptrPtrInt = &ptrInt;
   cout << "Give m and n"; cin >> m >> n;
   ptrInt = &n; cout << *(*ptrPtrInt) << endl;
   ptrInt = &m; cout << *(*ptrPtrInt) << endl;
   return 0;
}
```

Content of (0x740)

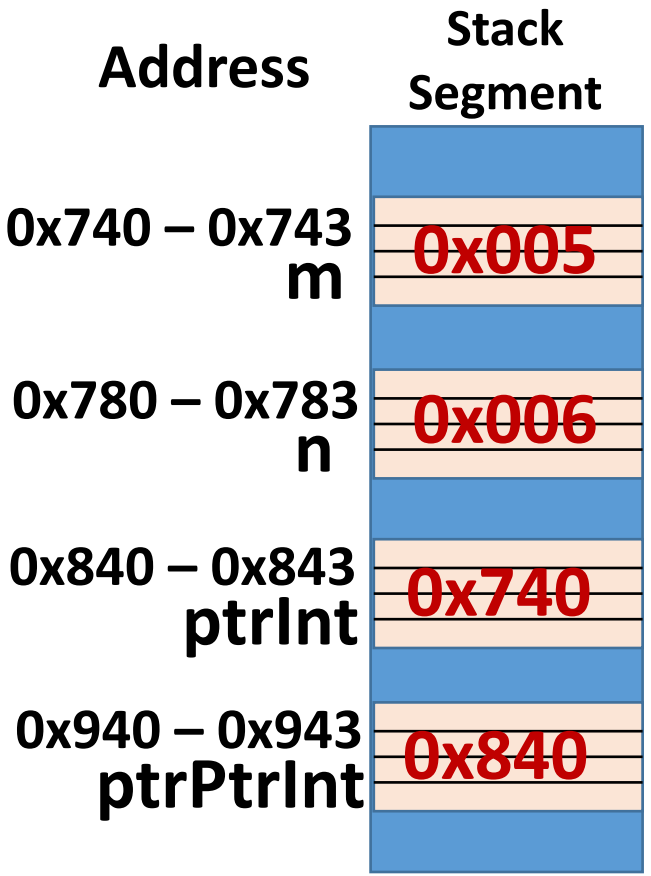


A C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n;
   int * ptrInt;
   int ** ptrPtrInt;
   ptrPtrInt = &ptrInt;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; cout << *(*ptrPtrInt) << endl;
   ptrInt = &m; cout << *(*ptrPtrInt) << endl;
   return 0;
}
```

Print 5



Memory Update using "*" Operator

IIT Bombay

- So far, we've used expressions like "* ptrA" to **read** the contents of memory at address given by ptrA
- We can also use "* ptrA" to **write** the contents of memory at address given by ptrA

*ptrA = b + c;

stores the value of expression "b + c" as the new content of memory at address given by ptrA

Another C++ Program With Pointers

IIT Bombay

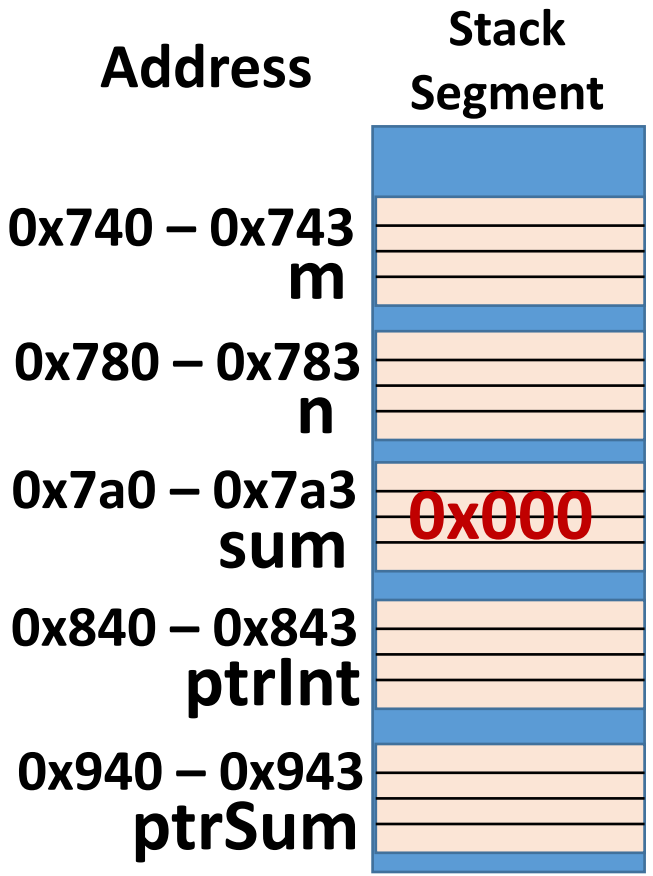
```
int main()
{  int m, n, sum = 0;
   int * ptrInt;
   int * ptrSum;
   ptrSum = &sum;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; *ptrSum += *ptrInt;
   ptrInt = &m;  *ptrSum += *ptrInt;
   cout << "Sum: " << sum << endl;
   return 0;
}
```

If we give values of 5 and 6 for m and n, what does the program output?

Another C++ Program With Pointers

IIT Bombay

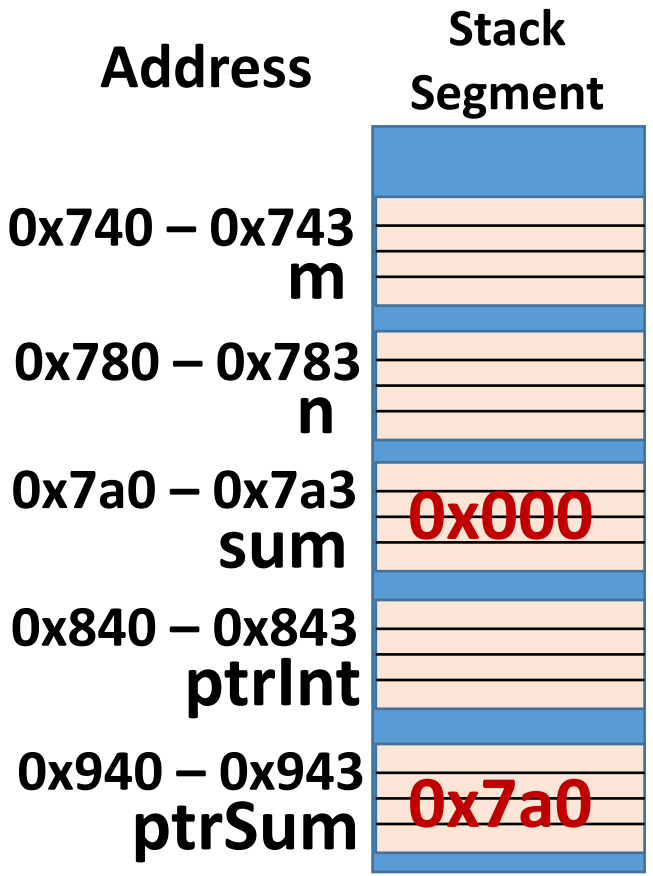
```
int main()
{  int m, n, sum = 0;
   int * ptrInt;
   int * ptrSum;
   ptrSum = &sum;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; *ptrSum += *ptrInt;
   ptrInt = &m;  *ptrSum += *ptrInt;
   cout << "Sum: " << sum << endl;
   return 0;
}
```



Another C++ Program With Pointers

IIT Bombay

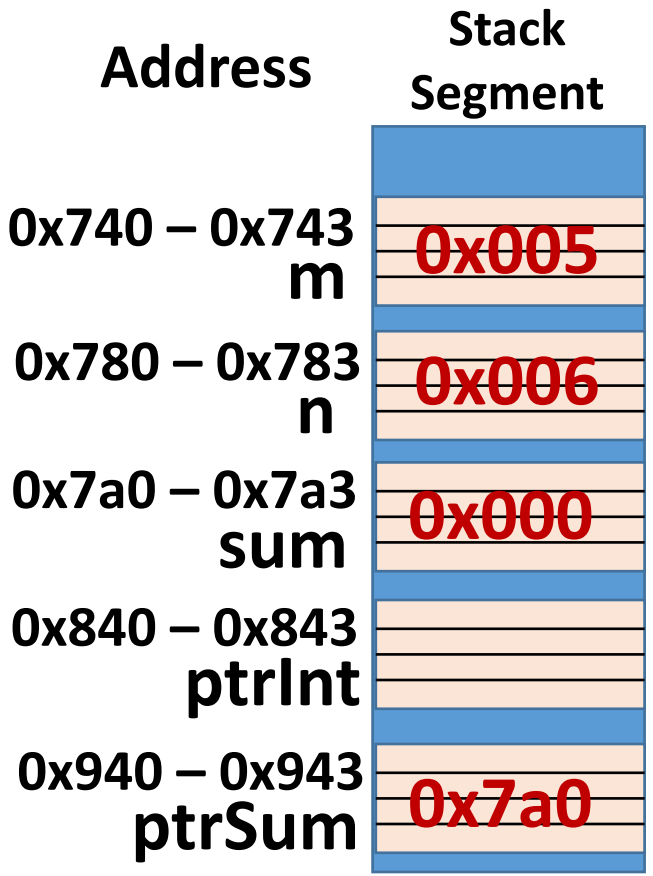
```
int main()
{  int m, n, sum = 0;
   int * ptrInt;
   int * ptrSum;
   ptrSum = &sum;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; *ptrSum += *ptrInt;
   ptrInt = &m;  *ptrSum += *ptrInt;
   cout << "Sum: " << sum << endl;
   return 0;
}
```



Another C++ Program With Pointers

IIT Bombay

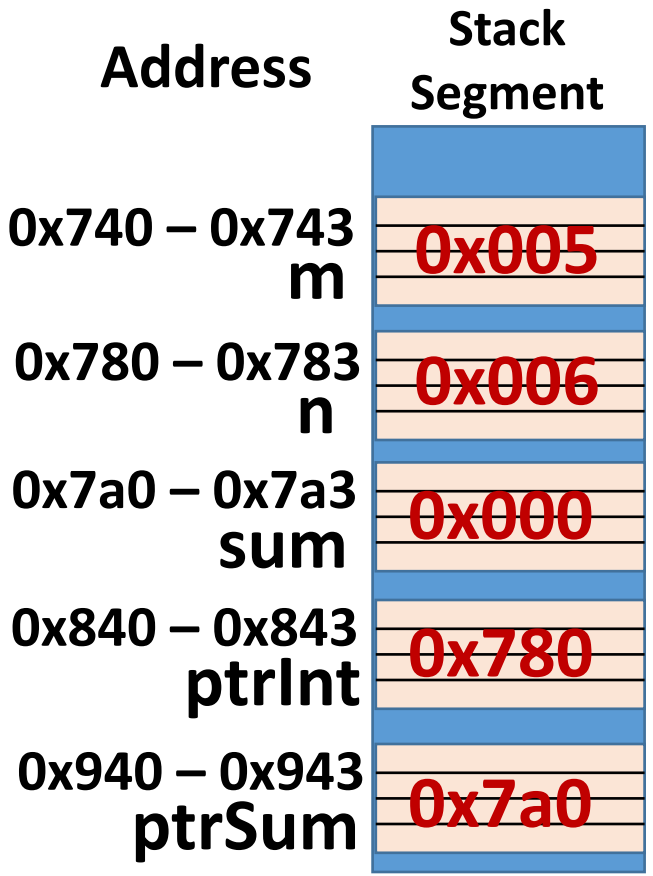
```
int main()
{  int m, n, sum = 0;
   int * ptrInt;
   int * ptrSum;
   ptrSum = &sum;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; *ptrSum += *ptrInt;
   ptrInt = &m;  *ptrSum += *ptrInt;
   cout << "Sum: " << sum << endl;
   return 0;
}
```



Another C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n, sum = 0;
   int * ptrInt;
   int * ptrSum;
   ptrSum = &sum;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; *ptrSum += *ptrInt;
   ptrInt = &m;  *ptrSum += *ptrInt;
   cout << "Sum: " << sum << endl;
   return 0;
}
```

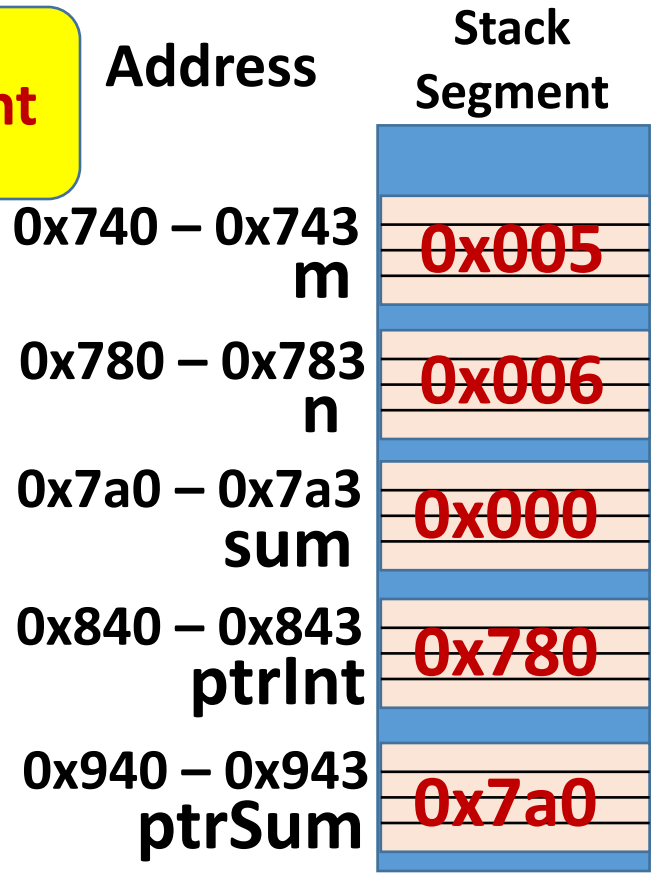


Another C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n, sum = 0;
   int * ptrInt;
   int * ptrSum;
   ptrSum = &sum;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; *ptrSum += *ptrInt;
   ptrInt = &m;  *ptrSum += *ptrInt;
   cout << "Sum: " << sum << endl;
   return 0;
}
```

*ptrSum = *ptrSum + *ptrInt

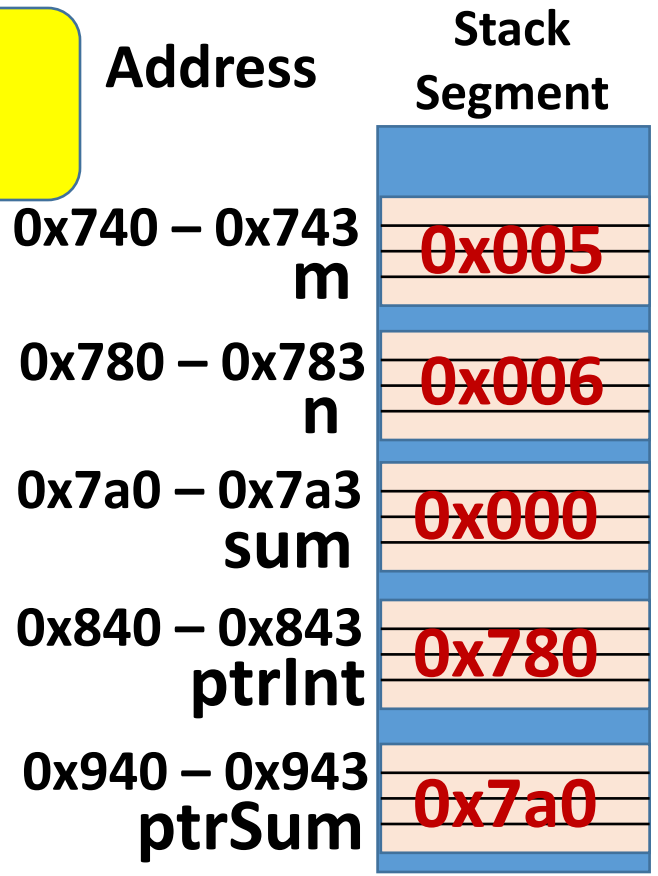


Another C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n, sum = 0;
   int * ptrInt;
   int * ptrSum;
   ptrSum = &sum;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; *ptrSum += *ptrInt;
   ptrInt = &m;  *ptrSum += *ptrInt;
   cout << "Sum: " << sum << endl;
   return 0;
}
```

*ptrSum = 0x000 + 0x006

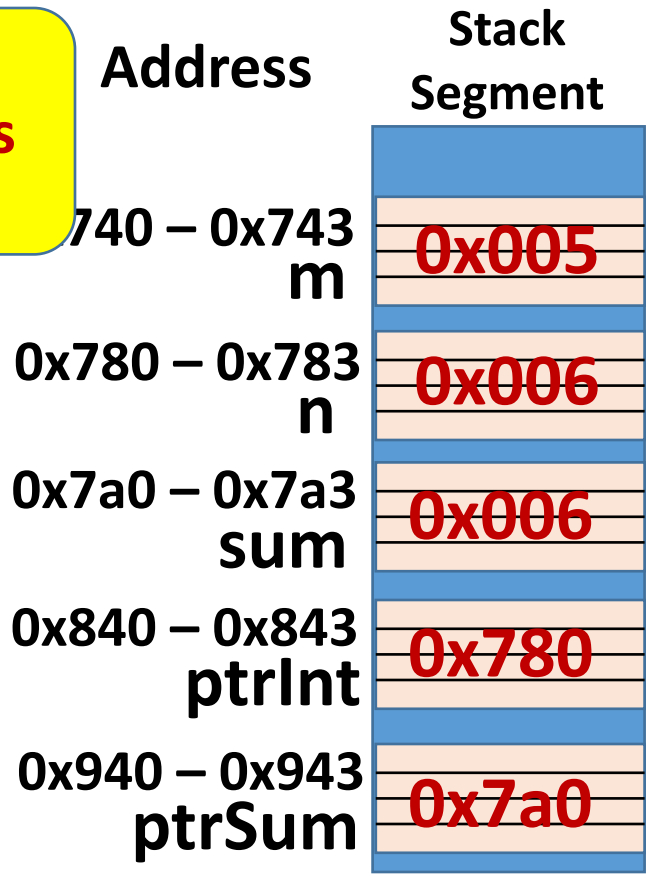


Another C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n, sum = 0;
   int * ptrInt;
   int * ptrSum;
   ptrSum = &sum;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; *ptrSum += *ptrInt;
   ptrInt = &m;  *ptrSum += *ptrInt;
   cout << "Sum: " << sum << endl;
   return 0;
}
```

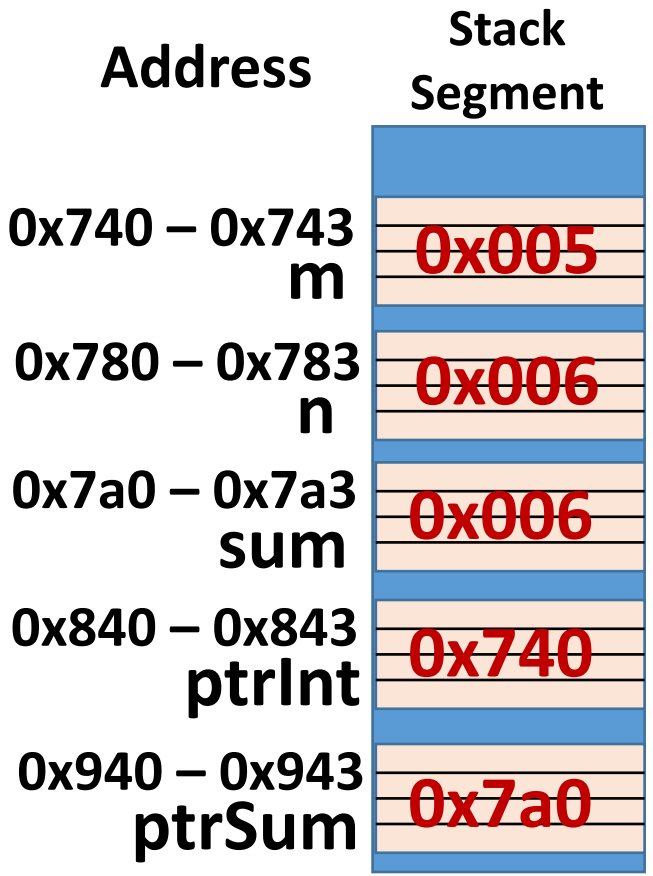
*ptrSum = 0x006
Update contents at address 0x7a0 to 0x006



Another C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n, sum = 0;
   int * ptrInt;
   int * ptrSum;
   ptrSum = &sum;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; *ptrSum += *ptrInt;
   ptrInt = &m;  *ptrSum += *ptrInt;
   cout << "Sum: " << sum << endl;
   return 0;
}
```

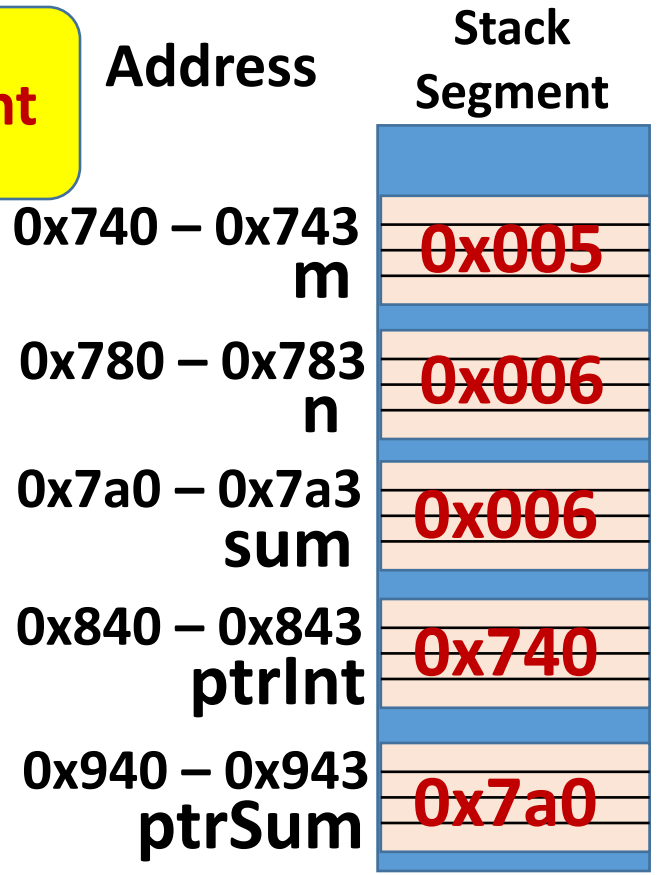


Another C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n, sum = 0;
   int * ptrInt;
   int * ptrSum;
   ptrSum = &sum;
   cout << "Give m and n:    cin >> m >> n;
   ptrInt = &n; *ptrSum +=  *ptrInt;
   ptrInt = &m; *ptrSum += *ptrInt;
   cout << "Sum: " << sum << endl;
   return 0;
}
```

*ptrSum = *ptrSum + *ptrInt

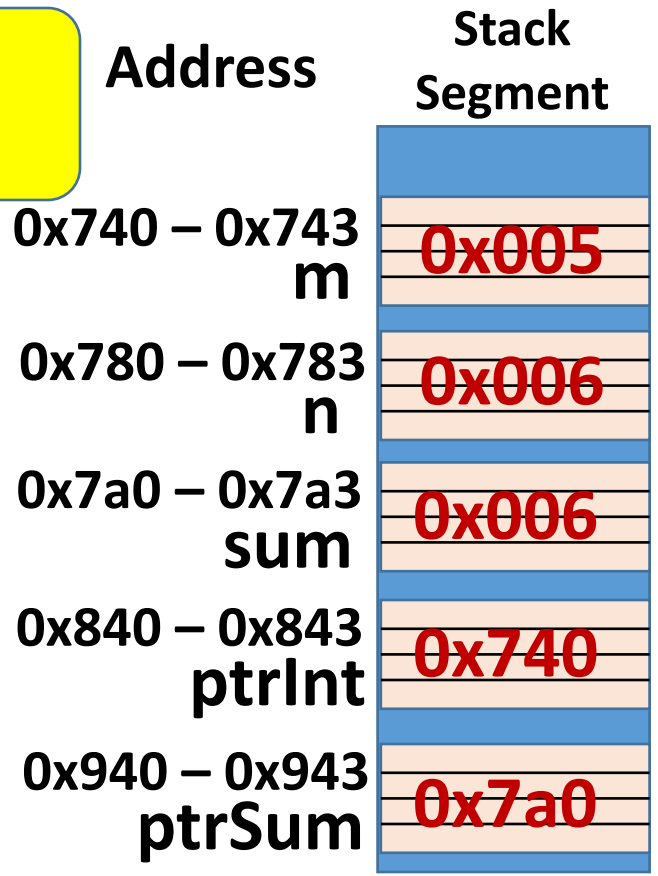


Another C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n, sum = 0;
   int * ptrInt;
   int * ptrSum;
   ptrSum = &sum;
   cout << "Give m and n: "; cin >> m >> n;
   ptrInt = &n; *ptrSum += *ptrInt;
   ptrInt = &m; *ptrSum += *ptrInt;
   cout << "Sum: " << sum << endl;
   return 0;
}
```

*ptrSum = 0x006 + 0x005

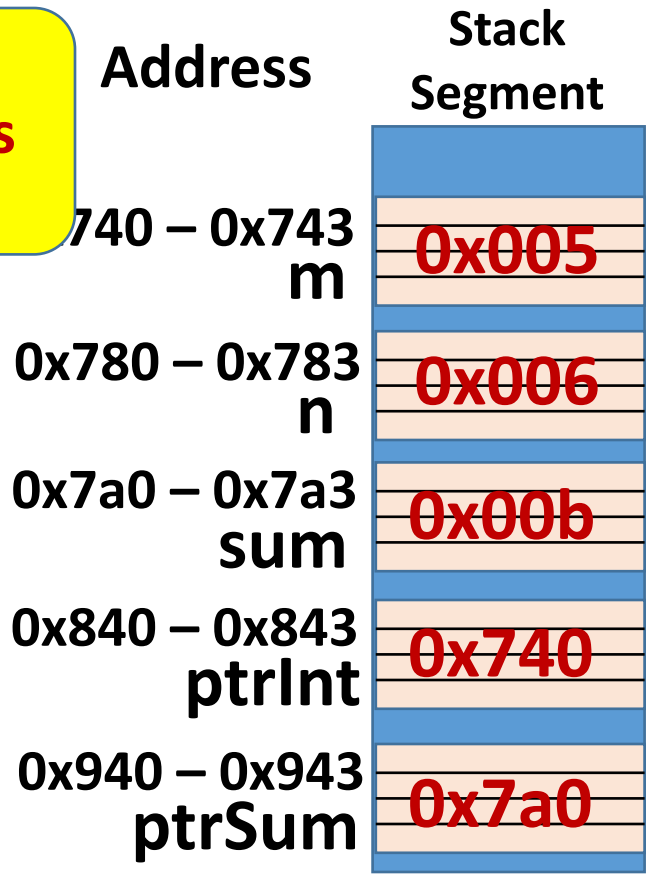


Another C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n, sum = 0;
   int * ptrInt;
   int * ptrSum;
   ptrSum = &sum;
   cout << "Give m and n:    cin >> m >> n;
   ptrInt = &n; *ptrSum +=  *ptrInt;
   ptrInt = &m; *ptrSum += *ptrInt;
   cout << "Sum: " << sum << endl;
   return 0;
}
```

***ptrSum = 0x00b**
Update contents at address 0x7a0 to 0x00b

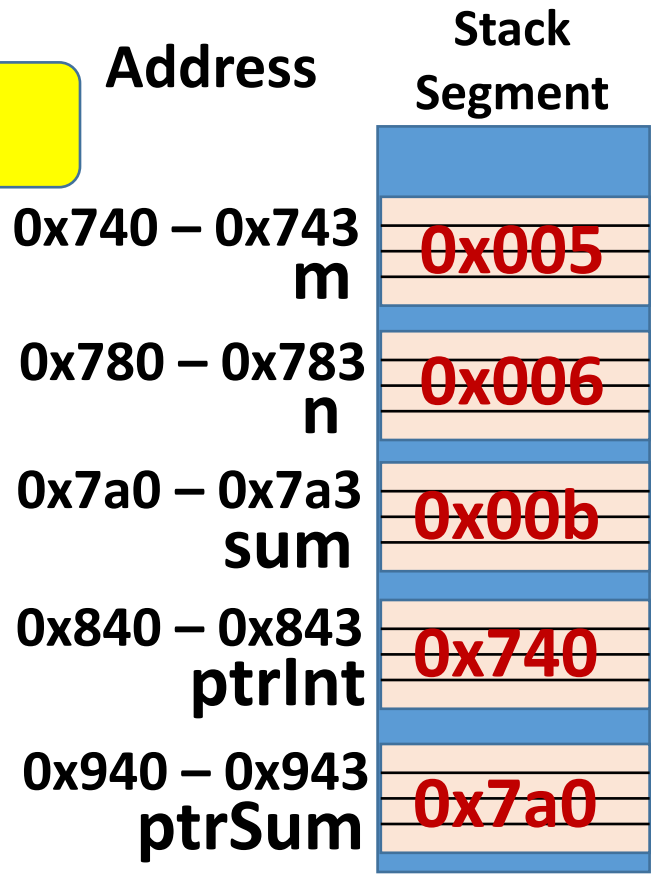


Another C++ Program With Pointers

IIT Bombay

```
int main()
{  int m, n, sum = 0;
   int * ptrInt;
   int * ptrSum;
   ptrSum = &sum;
   cout << "Give m and n"; cin >> m >> n;
   ptrInt = &n; *ptrSum += *ptrInt;
   ptrInt = &m;  *ptrSum += *ptrInt;
   cout << "Sum: " << sum << endl;
   return 0;
}
```

Print "Sum: 11"



Summary

- Use of “address of” and “content of” operators in C++
  - Unary & and unary * operators
- Understanding how contents of memory locations change when executing programs with pointers
- Using “content of” operator to update memory locations