Online Judge and C

Roy Chan

CSC2100B Data Structures Tutorial 1

January 12, 2009

Information

Your TA team

- CHAN Kai Chi
  - Email: kcchan[AT]cse.cuhk.edu.hk
- FUNG Wai Shing
  - Email: wsfung[AT]cse.cuhk.edu.hk
- MA Hao
  - Email: hma[AT]cse.cuhk.edu.hk

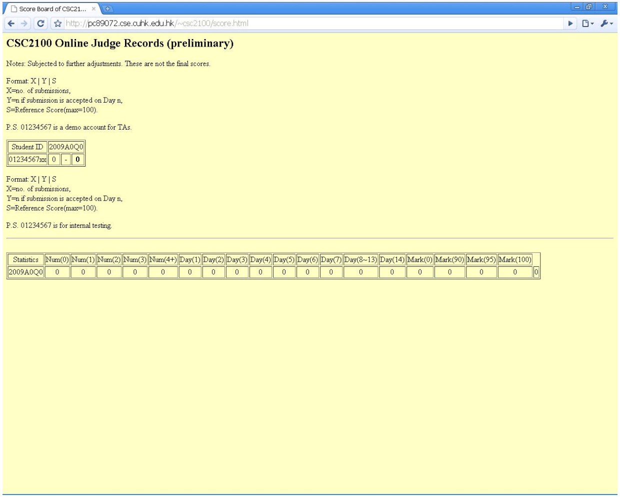
Course Information

- Course Web Page
  - http://wiki.cse.cuhk.edu.hk/irwin.king/teaching/csc2100b/2009
- Course Newsgroup
  - news://news.erg.cuhk.edu.hk/cuhk.cse.csc2100b
- Anti-plagiarism Policy
  - http://www.cuhk.edu.hk/policy/academichonesty

Assignment

- There will be both written and programming parts in assignments.
  - Written part: submit to the assignment box in 10/F SHB.
  - Programming part: via Online Judge systems.
- You will receive your login Id for CSC2100B online judge via your sxxxxxxx@mailserv.cuhk.edu.hk email account.
  - Keep it safe and do not disclose it.

Online Judge



Score Board of CSC21...

http://pc89072.cse.cuhk.edu.hk/~csc2100/score.html

CSC2100 Online Judge Records (preliminary)

Notes: Subjected to further adjustments. These are not the final scores.

Format: X | Y | S
X=no. of submissions,
Y=n if submission is accepted on Day n,
S=Reference Score(max=100).

P.S. 01234567 is a demo account for TAs.

Student ID	2009A0Q0		
01234567xx	0	-	**0**

Format: X | Y | S
X=no. of submissions,
Y=n if submission is accepted on Day n,
S=Reference Score(max=100).

P.S. 01234567 is for internal testing.

Statistics	Num(0)	Num(1)	Num(2)	Num(3)	Num(4+)	Day(1)	Day(2)	Day(3)	Day(4)	Day(5)	Day(6)	Day(7)	Day(8~13)	Day(14)	Mark(0)	Mark(90)	Mark(95)	Mark(100)	
2009A0Q0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Writing Your Assignment Program

- Write your program using your favorite editor, e.g., vi, vim, pico, emacs
- Add a header at the first line of your program

```
/* CSC2100@ 09123456 a1234567 assg0_question0 */
#include <stdio.h>

int main(int argc, char **argv)
{
 int a, b;

 printf("Please enter two numbers: \n");
 scanf("%d %d", &a, &b);
 printf("The numbers you entered are %d and %d \n", a, b);
 printf("And their sum is %d \n", a + b);

 return 0;
}
```

/* CSC2100@ 09123456 a1234567 assg0_question0 */

1. Course Code: CSC2100@
2. Student ID: 09123456
3. Login ID: a1234567
4. Problem ID: assg0_question0

Submitting Your Program

- Compile and test your program in Unix
  - To compile in Unix:

    ```
    gcc -o myprogram myprogram.c
    ```
  - To run:

    ```
    ./myprogram
    ```
- Submit to Online Judge when ready
  - Login to a sparc machine
  - `mail csc2100@pc89072 < myprogram.c`
- Wait for reply in your CSE mailbox

Online Judge Reply Messages

- Accepted
  - Congratulation
- Submission Error
  - Check your header line
- Compile Error
- Runtime Error
  - Time Limit Exceeded
    - Check your algorithm. Infinite loops?
  - Output Limited Exceeded
    - Check your algorithm. Infinite loops?
  - Float Point Exceptions
    - Division by 0
  - Segmentation Fault, Bus Error, ...
    - Check your code, e.g. errors in pointers.
- Wrong Answer.
- Presentation Error.

A word about ”Presentation Error”

- Make sure your output is EXACTLY the same as the specification (or sample program, if provided).
    - Check for upper case / lower case letters.
    - Check for extra spaces / extra blank lines.
        - There should be no extra spaces at the end of a line, and no extra blank lines at the end of the outputs.

Demostration

Question Id: 2009A0Q0
Write a program to calculate the products of two integers.

- Input the number of products to be computed.
- While the number of products is not exceeded, do:
  - Input two integers
  - Display their product
- You can assume the inputs are within the range $0 < x < 10000$.

Sample input

```
3
1 2
3 4
5 6
```

Expected output

```
P: 2
P: 12
P: 30
```

Attemp #1

```
/* CSC2100@ 01234567xx aaaaaaaaxx WRONG-QID */
#include <stdio.h>

int main(int argc, char **argv){
  short a = 0;
  short b = 0;
  short p;
  int i=0;

  int count;
  scanf("%d", &count);
  while (i < count) {
    scanf("%hd %hd", &a, &b);
    p=a*b;
    printf("P:  %hd \n", p);
  }

  return 0;
}
```

Submission Error: Please check the header format of your program.

Attemp #2

```
/* CSC2100@ 01234567xx aaaaaaaaxx 2009A0Q0 */
#include <stdio.h>

int main(int argc, char **argv){
  short a = 0;
  short b = 0;
  short p;
  int i=0;

  int count;
  scanf("%d", &count);
  while (i < count) {
    scanf("%hd %hd", &a, &b);
    p=a*b;
    printf("P:  %hd \n", p);
  }

  return 0;
}
```

Runtime Error: Time Limit Exceeded.

Attemp #3

```
/* CSC2100@ 01234567xx aaaaaaaaxx 2009A0Q0 */
#include <stdio.h>

int main(int argc, char **argv){
  short a = 0;
  short b = 0;
  short p;
  int i=0;

  int count;
  scanf("%d", &count);
  while (i < count) {
    scanf("%hd %hd", &a, &b);
    p=a*b;
    printf("P:  %hd \n", p);
    i++;
  }

  return 0;
}
```

Wrong Answer: Please check your program.

Attemp #4

```
/* CSC2100@ 01234567xx aaaaaaaaxx 2009A0Q0 */
#include <stdio.h>

int main(int argc, char **argv){
  short a = 0;
  short b = 0;
  int p;
  int i=0;

  int count;
  scanf("%d", &count);
  while (i < count) {
    scanf("%hd %hd", &a, &b);
    p=a*b;
    printf("P:  %d \n", p);
    i++;
  }

  return 0;
}
```

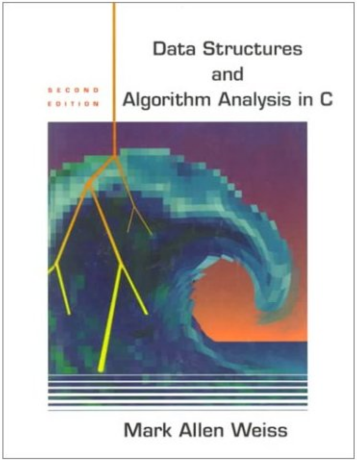
Presentation Error

Attemp #5

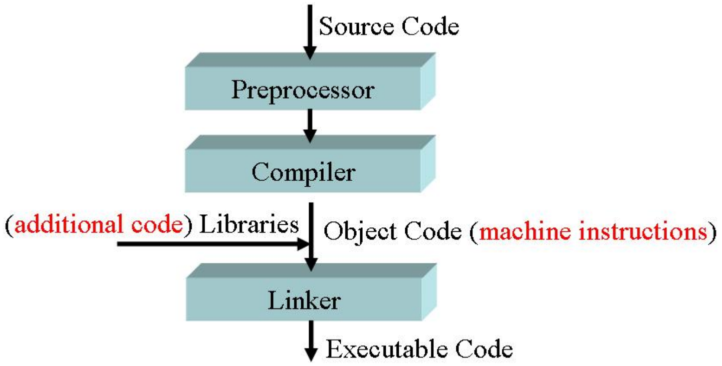
```
/* CSC2100@ 01234567xx aaaaaaaaxx 2009A0Q0 */
#include <stdio.h>

int main(int argc, char **argv){
  short a = 0;
  short b = 0;
  int p;
  int i=0;

  int count;
  scanf("%d", &count);
  while (i < count) {
    scanf("%hd %hd", &a, &b);
    p=a*b;
    printf("P: %d\n", p);
    i++;
  }

  return 0;
}
```

Accepted: Congratulations!

Introduction to C



The C Compilation Model


Source Code
Preprocessor
Compiler
(additional code) Libraries
Object Code (machine instructions)
Linker
Executable Code

C Operators

- Arithmetic:

  `+,-,*,/,%,++,--`

  ```
  int a = 10, b, c;
  b = a++; /* a is now 11, b is 10 */
  c = ++b; /* a, b, c are all 11 */
  ```

- Assignment:

  `=,+=,-=,*=,/=,%=`

  ```
  x += 2;
  x %= 2;
  ```

- Relational:

  `>,<,>=,<=,==,!=`

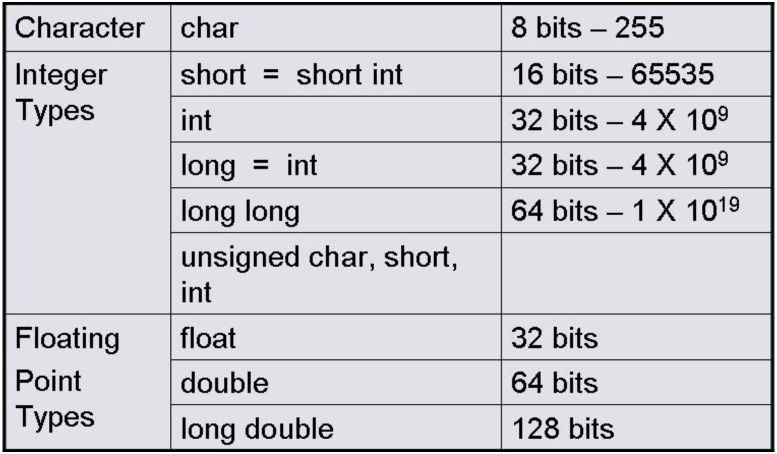
- Logical:

  `&&,||,!`

- Bitwise:

  `<<,>>,&,^,|`

Size



Character	char	8 bits – 255
Integer Types	short = short int	16 bits – 65535
	int	32 bits – 4 X 10^9
	long = int	32 bits – 4 X 10^9
	long long	64 bits – 1 X 10^{19}
	unsigned char, short, int	
Floating Point Types	float	32 bits
	double	64 bits
	long double	128 bits

Notes

- There is no boolean type in C.
  - Instead, non-zero values mean "true", zero means "false".

    ```
    int i = 5;
    while (i) {
      printf("%d \n", i);
      i--;
    }
    ```
- No class and subclasses, no methods, no interfaces.
  - Think of everything belongs to the same class.
- No public / private / protected...
- Instead, we have functions, pointers, structures, and dynamic memory allocations.

Constants

Constants can be defined using **#define** at beginning of file.

```
/* CSC2100@ 09123456 a1234567 assg0_question0 */
#include <stdio.h>
#define PI 3.14159

int main(int argc, char **argv)
{
  float radius, area;

   radius = 5;
   area = radius * radius * PI;

   printf("The area is %f \n", area);

  return 0;
}
```

Functions

```
#include <stdio.h>

int sum(int x, int y){
        return x + y;
}

int main(int argc, char **argv){
  int a, b;
  printf("Enter 2 numbers");
  scanf("%d %d", &a, &b);
  printf("%d", sum(a,  b));
  return 0;
}
```

Notes In stardard ANSI C:

- Local variables should be declared at the beginning of the function.
- Functions should be defined or declared before they are used.
- Local variables will not be automatically initialized.
  - E.g, "int a" may contain garbage until a value is assigned.

Pointer Variables

- Pointer variables are variables that store memory addresses.
- Pointer Declaration:

```
int x, y = 5;
int *ptr;
/*ptr is a POINTER to  an integer variable*/
```

- Reference operator &:

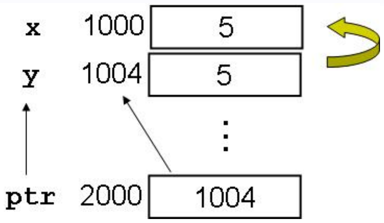
```
ptr = &y;
/*assign ptr to the MEMORY ADDRESS of y.*/
```

- Dereference operator *:

```
x = *ptr;
/*assign x to the int that is pointed to by ptr */
```

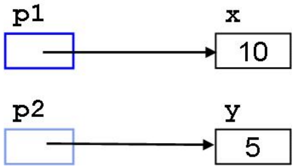
Pointer Example 1

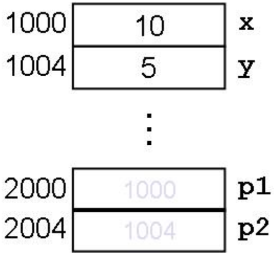
```
int x;
int y = 5;
int *ptr;

ptr = &y;

x = *ptr;
```



Pointer Example 2

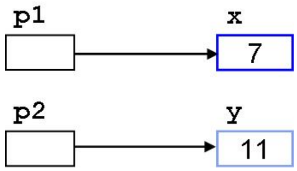
```
int x = 10, y = 5;
int *p1, *p2;
p1 = &x;
p2 = &y;
```

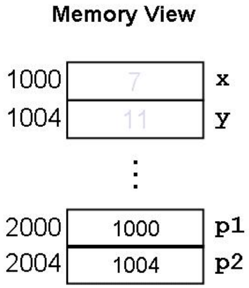




Pointer Example 2

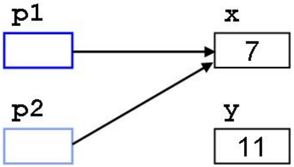
```
*p1 = 7;
*p2 = 11;
```

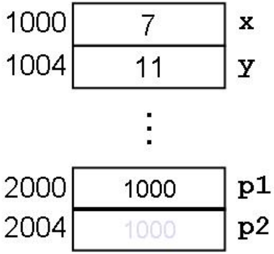




Pointer Example 2

```
p2 = p1;  // Not the same as *p2 = *p1
```





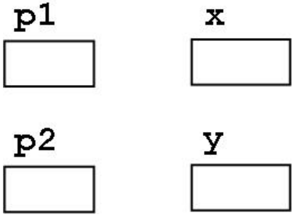
Pointer Example 3: A function that swaps two variables

```
void swap (int *px, int *py) {
  int temp;
  temp = *px;
  *px = *py;
  *py = temp;
}

int main() {
  int x=1; y=2;
  swap(&x, &y);
  return 0;
}

//Note: this is not possible in Java!
```

Pointer Exercise 1

What will be the value of x, y, *p1 and *p2?

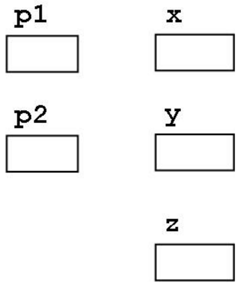
```
int x = 7, y = 11;
int *p1, *p2;
p1 = &x;
p2 = &y;
*p1 = y;
*p2 = x;
```



Pointer Exercise 2

What will be the value of x, y, *p1 and *p2?

```
int x = 7, y = 11, z = 3, *p1, *p2;
p2 = &x;
p2 = &y;

*p2 = 5;
p1 = p2;
p2 = &z;
y = 6;

z = *p1;
*p2 = x;
```



Structure

- A collection of values (members)
  - Like a class in java or C++, but without methods.

```
struct time {
  int hh;
  int mm;
  int ss;
};
...
struct time t1;
t1.hh=20;
t1.mm=12;
t1.ss=30;
...
```

Structure

- We can also use pointer to structure.

```
struct time {
  int hh;
  int mm;
  int ss;
};
struct time *t1;
(*t1).hh=20;
```

- Pointer to structure is very common, so we gave it a short hand. The above is equivalent to:

```
struct time *t1;
t1->hh=20;  /* Same as (*t1).hh=20;    */
```

Structure

- Allow us to define alias for a data type.

```
typedef int My_integer_type;
...
My_integer_type x=3;
```

- Typedef can be used for structures.

```
typedef struct {
   int hh;
   int mm;
   int ss
} Time_type;
...
Time_type t1;
T1.hh=12;
...
```

Question