CSE220 - Programming in C
Fall 2015

• Midterm Review
Midterm Exam

• **Wednesday (Oct 14)**
  • in-class, 50 minutes
  • 5 problems, 15% of the total grade
  • closed-book, but you can bring a piece of paper with whatever information you would like to write down
  • bring a calculator if you wish
Example (I): Base Conversion

Convert the numbers into the asked base format:

Convert Decimal to Octal

\[(74)_{10} = (?)_{8}\]
Example (I)

Decimal to Octal Representation

Step (I): \((74)_{10} = (1001010)_2\)

Step (ii): \((1001010)_2 = 112_8\)
Example (I)

\[(1010 \ 0000 \ 0000 \ 0000) = (?)_{16}\]

\[
\begin{array}{cccc}
10 & 0 & 0 & 0 \\
A & 0 & 0 & 0 \\
\end{array}
\]

- In hexadecimal representation, we use A ~ F to represent 10 ~ 15
Example (I)

- Convert from Hexadecimal to Decimal
  \[0xA123 = (?)_{10}\]

\[
0xA123 = 3 \times 16^0 + 2 \times 16^1 + 1 \times 16^2 + \textcolor{red}{?} \times 16^3
\]

\[
= 3 \times 16^0 + 2 \times 16^1 + 1 \times 16^2 + 10 \times 16^3
\]

\[
= 41251
\]
Example (II)  Bitwise Operator

if second and seventh bits are zero or not?

input: \( x \rightarrow 1010 \ 1111 \)

Since we are interested in those two positions, we are going to isolate those two bits first. So, we have the following mask:

\[
\text{mask1} = 0100 \ 0010
\]

Step 1) To isolate we are going to use ‘\&’ operator:

\[
y = x \& \text{mask1}
\]

\[
x = 1010 \ 1111
\]

\[
\text{mask1}= 0100 \ 0010
\]

\[
y = 00000010
\]

Step 2) Then we should check if the value of those bits are zero or not. So we define a new mask as following:

\[
\text{mask2} = 0000 \ 0000
\]

\[
z = y \oplus \text{mask2}
\]

\[
y = 00000010
\]

\[
\text{mask2}= 0000 \ 0000
\]

\[
z = 00000010
\]

Now

if \( z = 0 \) => Both those two positions are zero.
if \( z \neq 0 \) => Both those two positions are NOT zero.
Example (III): read and sum 10 numbers

```c
#include <stdio.h>
main()
{
    int i, a, sum;
    sum = 0;
    for (i = 0; i < 10; i++)
    {
        Body
    }
    printf("total is %d", sum);
}
```
Example (III): read and sum 10 numbers

```c
#include <stdio.h>
main()
{
    int i, a, sum;
    sum = 0;
    for (i = 0; i < 10; i++)
    {
        printf("enter number \n");
        scanf("%d", &a);
        sum = sum + a;
    }
    printf("total is %d", sum);
}
```
Example (III): largest of 10 numbers

we want to read 10 numbers and find the maximum among them
Example (III): largest of 10 numbers

we want to read 10 numbers and find the maximum among them

```c
#include <stdio.h>
int main()
{
    int i, a, max;
    printf("enter number \n");
    scanf("%d", &a);
    max = a;
    for (i=1; i < 10; i++)
    {
        printf("enter number \n");
        scanf("%d", &a);
        if (a > max)
            max = a;
    }
    printf("largest is %d", max);
}
```
Example (III)

• Write a program that given n values, find the smallest value among them
• For example:

\[
\begin{align*}
n &= 6 \\
\text{inputs: } 6, 12, 1, 97, 5, 27 \\
\text{output: } 1
\end{align*}
\]
Example(III)
Example(III)

```c
#include<stdio.h>

int main()
{
    int n = 0, min = 0, current;

    printf("Enter the number of given numbers: \n");
    scanf("%d", &n);

    return 0;
}
```
Example (III)

```c
#include <stdio.h>

int main()
{
    int n = 0, min = 0, current;

    printf("Enter the number of given numbers: \n");
    scanf("%d", &n);

    printf("Enter the numbers: \n");
    scanf("%d", &current);

    min = current;

    return 0;
}
```
Example (III)

```c
#include<stdio.h>

int main()
{
    int n = 0, min = 0, current;

    printf("Enter the number of given numbers: \n");
    scanf("%d", &n);

    printf("Enter the numbers: \n");
    scanf("%d", &current);

    min = current;
    for (int i = 2; i <= n; i++)
    {
        /* read the next number and compare to minimum so far */
    }

    printf("The minimum number is: %d\n", min);
    return 0;
}
```
```c
#include<stdio.h>

int main()
{
    int n = 0, min = 0, current;

    printf("Enter the number of given numbers: \n");
    scanf("%d", &n);

    printf("Enter the numbers: \n");
    scanf("%d", &current);

    min = current;
    for (int i = 2; i <= n; i++)
    {
        /* read the next number and compare to minimum so far */
        scanf("%d", &current);

        if (current < min)
            min = current;
    }

    printf("The minimum number is: %d\n", min);
    return 0;
}
```