Basis and Practice in Programming
week4
Character Representation (1/2)

• Character representation in C
  – Just integer number
  – ASCII : American Standard Code for Information Interchange

```c
/* practice 1 : character representation */
#include <stdio.h>

int main(void)
{
  char sa = 'a', sb = 'b';
  char la = 'A', lb = 'B';

  printf("%c %c %c %c\n", sa, sb, la, lb);
  printf("%d %d %d %d\n", sa, sb, la, lb);

  return 0;
}
```

**Printing format**

- `%c` : character format
- `%d` : integer format
Character Representation (2/2)

- Character representation in C (continued)

```c
/* practice 2 : integer to character*/
#include <stdio.h>

int main(void)
{
    char foo = 48, bar = 97;
    int i = 0;

    while (i < 10) {
        foo += 1;
        bar += 1;
        printf("%c %c\n", foo, bar);
        i++;
    }

    return 0;
}
```

ASCII Table: [http://www.asciitable.com/]
Integer Representation (1/3)

• Integer representation in C
  – Char type: 8 bits (1 byte)
  – Int type: 32 bits (4 bytes)
  – Sign bit: representing +/-, Value bits: value of number
Integer Representation (2/3)

- Integer representation in C (continued)

<table>
<thead>
<tr>
<th>Data type</th>
<th>Memory size</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>char (signed)</td>
<td>1 byte</td>
<td>-128 ~ 127</td>
</tr>
<tr>
<td>char (unsigned)</td>
<td>1 byte</td>
<td>0 ~ 255</td>
</tr>
<tr>
<td>int (short)</td>
<td>2 bytes</td>
<td>-32768 ~ 32767</td>
</tr>
<tr>
<td>int (signed)</td>
<td>4 bytes</td>
<td>-2147483648 ~ 2147483647</td>
</tr>
<tr>
<td>int (unsigned)</td>
<td>4 bytes</td>
<td>0 ~ 4294967295</td>
</tr>
</tbody>
</table>
Integer Representation (3/3)

- Integer representation in C (continued)

```c
/* practice 3 : integer representation*/
#include <stdio.h>

int main(void)
{
    int num = 0xffffffff; //maximum value in int
    printf("%d %u\n", num, num);
    return 0;
}
```

Printing format
%u : unsigned integer format
Printing Format (1/2)

- C supports formatted printing

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%c</td>
<td>Single character</td>
</tr>
<tr>
<td>%d (=%i) / %u</td>
<td>Signed integer / Unsigned integer</td>
</tr>
<tr>
<td>%f</td>
<td>Signed floating point number</td>
</tr>
<tr>
<td>%s</td>
<td>String</td>
</tr>
<tr>
<td>%0</td>
<td>Unsigned octal number</td>
</tr>
<tr>
<td>%x / %X</td>
<td>Unsigned hexadecimal (small/capital letter)</td>
</tr>
<tr>
<td>%e</td>
<td>e-notation of real number</td>
</tr>
<tr>
<td>%%</td>
<td>printing ‘%’</td>
</tr>
</tbody>
</table>
/ * practice 4 : printf exercise*/
#include <stdio.h>

int main(void)
{
    printf("Menu...
");
    printf("%10s: $%u
", "Spaghetti", 10);
    printf("%10s: $%2.6f
", "Pizza", 15.50);
    printf("%10s: $%2.6f
", "Salad", 6.50);
    printf("%10s: $%2.6f
", "Coke", 1.50);

    return 0;
}

If Statement (1/2)

• If-else statement
  – Control flow statement
  – Controlled by given condition

```c
if (a == 1)
    printf("a is 1");
else
    printf("a isn’t 1");
```
/* practice 5 : if statement exercise*/
#include <stdio.h>

int main(void)
{
    int a;
    scanf("%d", &a);

    if (a > 10) {
        printf("a is bigger than 10\n"); // condition is true
    } else {
        printf("a is smaller than 10\n"); // condition is false
    }

    return 0;
}
Exercise

• **Letter changer** (Due date: Today’s 11:59 PM)
  – Enter one character (only Alphabet can be input)
  – Translate small letter to capital letter
  – Translate capital letter to small letter
  – Print translated letter as output
  – Must use if-statement