C++ Programming(Exercise) 6

- Arne Kutzner
- Hanyang University / Seoul Korea

Pointers

```cpp
#include <iostream>
using namespace std;

int main()
{
    int october = 19;
    int* pt = &october;

    cout << "value: october = " << october << " *pt = "
        << *pt << " pt = " << pt << endl;
    cout << "address: october = " << &october << ", *pt = "
        << &*pt << ", pt = " << pt << endl;

    return 0;
}
```

Pointer Addition

```cpp
#include <iostream>
using namespace std;

int main( )
{
    double wages[3] = { 10000.0, 20000.0, 30000.0 };
    short stacks[3] = { 3,2,1 };

    double* pw = wages;
    short* ps = &stacks[0];

    cout << "pw = " << pw << ", *pw = " << *pw << endl;
    pw = pw + 1;

    cout << "add 1 to the pw pointer : " << endl;
    cout << "pw=1" << pw << ", *pw = " << *pw << "₩n₩n";

    cout << "ps = " << ps << ", *ps = " << *ps << endl;
    ps = ps + 1;

    cout << "add 1 to the ps pointer:₩n";
    cout << "ps = " << ps << ", *ps = " << *ps << "₩n₩n";
    cout << "access two elements with array notation₩n";

    cout << "stacks[0] = " << stacks[0] << ", stacks[1] = "
        << stacks[1] << endl;

    cout << "access two elements with array notation₩n";
    cout << "*stacks = " << *stacks << ", *(stacks + 1)"
        << *(stacks + 1) << endl;

    cout << sizeof(wages) << " = size of wages array₩n";
    cout << sizeof(pw) << " = size of pw pointer₩n";

    return 0;
}
```

C++ strings

- String: null terminated array of characters
  - null terminated: A character at the end of the array has the value 0(null).

New &Delete

```
#include <iostream>
using namespace std;

int main()
{
    double* p3 = new double[3]; //allocates memory
    p3[0] = 0.2;
    p3[1] = 0.5;
    p3[2] = 0.8;

    cout << "p3[1] is " << p3[1] << endl;
    p3 = p3 + 1;
    cout << "now p3[0] is " << p3[0] << " and "
        << "p3[1] is " << p3[1] << endl;
    p3 = p3 - 1;
    delete[] p3;

    return 0;

}
```

pointers to strings

```
#include <iostream>
#include <cstring>
using namespace std;

int main( )
{
    char animal[20] = "bear";
    const char* bird = "wren";
    char* ps;
    cout << animal << " and ";
    cout << bird << endl;

    cout << "Enter a kind of animal: ";
    cin >> animal;
    ps = animal;

    cout << ps << "s!₩n";
    cout << "Before using strcpy( ):₩n";
    cout << animal << " at " << (int*)animal << endl;
    cout << ps << " at " << (int*)ps << endl;

    cout << ps << "s!₩n";
    cout << "Before using strcpy( ):₩n";
    cout << animal << " at " << (int*)animal << endl;
    cout << ps << " at " << (int*)ps << endl;

    ps = new char[strlen(animal) + 1];

    strcpy(ps, animal);
    cout << "After using strcpy( ):₩n";
    cout << animal << " at " << (int*)animal << endl;
    cout << ps << " at " << (int*)ps << endl;

    return 0;
}
```

strcat & strcmp

```cpp
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char str1[20] = "abcde";
    char str2[] = "fghij";

    strcat_s(str1, str2);

    if (strcmp(str1, "abcdefghij") == 0)
        cout << "str1 and ₩"abcdefghif₩" are identical." << endl;
    if (strcmp("123456", str1) != 0)
        cout << "₩"123456₩" and str1 are NOT identical. " << endl;
    return 0;
}
```

HW Solution 2-1

```
#include <iostream>
using namespace std;

int fac(int x)
{
    if (x <= 0)
        return 1;
    else
        return x * fac(x - 1);
}
int bc(int n, int r)
{
    return fac(n) / (fac(n - r) * fac(r));
}
void main()
{
    int n, r;
    std::cout << "Enter two natural numbers: ";
    std::cin >> n >> r;
    if (n < 0 && r < 0)
    {
        do
        {
            std::cout << "Please again enter two natural numbers !! " << std::endl;
            std::cout << "Enter two natrural number ";
            std::cin >> n >> r;
        } while (n < 0 || r < 0);
    }
    std::cout << "C(" << n << "," << r << ") = " << bc(n, r) << std::endl;
}
```

HW solution 2-2

```
#include <iostream>
using namespace std;

void permutNumbers(int data[], int x, int n)
{
    int k, temp;
    if (x == n - 1)
    {
        for (k = 0; k < n; k++)
            cout << data[k];
        if (k == n)
            cout << "  ";
    }
    else
    {
        for (k = x; k < n; k++)
        {
            temp = data[k];
            data[k] = data[x];
            data[x] = temp;
            permutNumbers(data, x + 1, n);
            temp = data[k];
            data[k] = data[x];
            data[x] = temp;
        }
    }
}
int main() {
    int a, i;
    int data[10];

    cout << "Enter one natural number : ";
    cin >> a;
    for (i = 0; i < a; i++)
        data[i] = i + 1;
    permutNumbers(data, 0, a);
    return 0;
}
```