**C++ 프로그래밍  
2019년 1학기 중간고사**

1. 다음 코드의 출력을 쓰시오.

#include<iostream>

int main(){

int i= 3;

while(i--){

int i = 100;

i--;

std::cout << i<< “ “;

}

return 0;

}

99, 99, 99

2. 프로그램에서 다음과 같은 2차원 배열이 선언되고, 이 배열이 메모리 주소(Memory Address) 0에서 시작한다고 가정하자.

int A[100][1000];

이때, A[10][10]의 주소를 10진수로 써라.

40040

3. 다음과 같이 두 개의 실수를 인자로 받아서 두 수의 합과 곱을 각각 sum 과 product 의 값으로 반환하는 함수 sum\_product ( ) 를 구현하라. 예를 들어,

int main ( ) {

double x, y, sum, product;

x = 2.718;

y = 3.142;

sum\_product (x, y, sum, product);

cout << sum << “ “ << product << endl;

}

의 출력은 다음과 같다.

5.86 8.539956

void sum\_product (double x, double y, double &sum, double &product)

{

sum = x + y;

product = x \* y;

}

4. 표준 출력 ( cin ) 으로부터 -20억과 +20억 사이의 두 정수를 한 번만 입력받아서 두 수 중 큰 수를 출력하는 c++ 소스 코드를 작성하라. g++ 를 이용하여 컴파일될 수 있게 헤더파일의 선언까지 포함하라. 입력 오류는 없다고 가정한다. 소스 코드는 공백 라인을 제외하고 12줄을 넘지 않아야 한다.

#include <iostream>

using namespace std;

int main(){

int a, b;

cin >> a >> b;

cout << ((a > b)? a : b) << endl;

return 0;

}

5. 다음 코드의 출력을 써라.

static int g = 10;

void proc(int num) {

static int g = 1;

g += num;

cout << ++g << endl;

}

int main(){

proc(g++);

proc(++g);

return 0;

}

12

25

6. n 개의 숫자를 가진 정수 배열에서, 주어진 값의 첫번째 위치 첨자를 반환하는 함수 find\_number( ) 를 구현하라. 주어진 값이 배열에 없으면 -1을 반환한다. 예를 들면 다음과 같은 main ( ) 함수에서 호출이 된다.

int main ( ) {

int a1[ ] = {0, 1, 2, 3};

int r;

r = find\_number(a1, 4, 100);

// 4개의 원소를 가진 a1 배열에서 100의 위치를 찾아서 반환하면 r의 값은 -1 이 되어야 한다.

r = find\_number(a1, 4, 2);

// 4개의 원소를 가진 a1 배열에서 2의 위치를 찾아서 반환하면 r의 값은2 이 되어야 한다.

}

int find\_number(int a[], int len, int key){

for (int i=0; i<len; i++){

if (a[i] == key) return i;

}

return -1;

}

7. 1부터 시작하는 n 개의 홀수로 이루어진 정수 배열을 생성하는 함수 void createOddArray ( ) 를 구현하고, main( ) 함수에서 호출할 때 빈칸을 채우시오. createOddArray ( ) 함수의 첫번째 인수는 함수 호출 후에 생성된 배열의 주소를 갖게 되고 두번째 인수는 생성될 배열의 원소 갯수 n 이다. 예를 들어 다음과 같이 n 이 10인 경우 함수 호출 후 배열 arr은 1 3 5 7 9 11 13 15 17 19 의 정수형 원소들로 구성된다.

int main() {

int \*arr = NULL;

createOddArray( , 10);

delete[] arr;

return 0;

}

|  |  |
| --- | --- |
| 답안 1 | 답안 2 |
| CreateOddArray(&arr, 10); | CreateOddArray(arr, 10); |
| void CreateOddArray(int\*\* ptr, int size)  {  \*ptr = new int[size]();  for (int i = 0, num = 0; i < size; ++num)  {  if (!(num % 2)) continue;  //(\*ptr)[i++] = num; or  \*(\*ptr + i++) = num;  }  } | void CreateOddArray(int \*&ptr, int size)  {  ptr = new int[size]();  for (int i = 0, num = 0; i < size; ++num)  {  if (!(num % 2)) continue;  //ptr[i++] = num; or  \*(ptr + i++) = num;  }  } |

8. 입력된 정수들에 따라 배열들을 동적으로 할당받아 다음과 같은 구조를 생성하는 프로그램을 구현하라. 입력 중 첫번째 숫자 n은 생성될 정수 포인터 배열 arr의 원소의 갯수를 나타낸다. arr 배열의 각 원소는 다시 동적으로 할당받은 정수 배열의 주소를 갖게 되는데 각 배열의 원소의 갯수는 n 이후에 순서대로 입력된 숫자와 같다. 그리고 각 배열의 원소들은 0부터 시작하여 1씩 증가하는 정수를 갖도록 한다.

즉 입력이 3 5 8 2 라면 3개의 원소를 갖는 정수 포인터 배열 arr이 먼저 만들어지고 arr 의 첫번째 원소는 5개의 정수로 이루어진 배열, 두번째 원소는 8개, 3번째 원소는 2개의 정수로 이루어진 배열이 되며 각각 다음과 같은 정수값들을 가진다.

그림에 보이는 배열들은 반드시 모두 동적으로 할당받아야 한다.

0

1

2

3

4

0

1

2

3

4

5

6

7

0

1

arr

#include <iostream>

using namespace std;

int main()

{

int \*\*arr;

int N; // the dimension of the 2D array

cin >> N;

arr = new int \*[N];

for(int n = 0; n < N; n++){ // get each 1D sub array's length & create 1D sub array

int subArrLen;

cin >> subArrLen;

arr[n] = new int[subArrLen];

for (int m = 0; m < subArrLen; m++) {

arr[n][m] = m;

cout << arr[n][m] << ' ';

}

cout << endl;

}

for(int n = 0; n < N; n++){

delete[] arr[n];

}

delete[] arr;

return 0;

}