**객체지향프로그래밍 LAB #05&06**

**<기초문제>\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. 아래의 등급 확인 프로그램을 작성하시오. (/\*구현\*/ 부분을 채울 것, 표의 상단: 소스코드, 하단: 실행결과) 단, if/else if/else만을 이용하여 구현하고, 아래의 조건에 맞게 구현하시오.  
- 0이상 100이하의 값이 아니면 점수가 잘못되었다고 출력  
- 90이상: A  
- 80이상 90미만: B  
- 70이상 80미만: C  
- 60이상 70미만: D  
- 60미만: F

|  |
| --- |
| #include<iostream>  using namespace std;  int main() {  char grade = 'X';  int score;  cout << "Enter your score: ";  cin >> score;  if (/\* 구현 \*/) {  /\* 구현 \*/  cout << "Your grade is " << grade << endl;  }  else {  cout << "The score (" << score << ") is invalid" << endl;  }  return 0;  } |
|  |
| #include<iostream>  using namespace std;  int main() {  char grade = 'X';  int score;  cout << "Enter your score: ";  cin >> score;  if (score >= 0 && score <= 100) {  if (score >= 90) {  grade = 'A';  }  else if (score >= 80) {  grade = 'B';  }  else if (score >= 70) {  grade = 'C';  }  else if (score >= 60) {  grade = 'D';  }  else {  grade = 'F';  }  cout << "Your grade is " << grade << endl;  }  else {  cout << "The score (" << score << ") is invalid" << endl;  }  return 0;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

2. while을 이용하여 2의 거듭제곱수를 표현하는 프로그램을 작성하시오. (/\*구현\*/ 부분을 채울 것) 단, setw값으로 10을 사용하시오.

|  |
| --- |
| #include<iostream>  #include<iomanip> // setw등 cout 표현을 멋지게  using namespace std;  int main() {  int num = 1;  int count = 1;  cout.imbue(locale("")); // 세 자리(1,000)마다 쉼표 찍기위함.  while (/\* 구현 \*/) {  /\* 구현 \*/  cout << count << '\t' << /\* 구현 \*/  /\* 구현 \*/  }  return 0;  } |
|  |
| #include<iostream>  #include<iomanip> // setw등 cout 표현을 멋지게  using namespace std;  int main() {  int num = 1;  int count = 1;  cout.imbue(locale("")); // 세 자리(1,000)마다 쉼표 찍기위함.  while (count <= 20) {  num \*= 2;  cout << count << '\t' << setw(10) << num << endl;  count++;  }  return 0;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

3. 단위 행렬 을 출력하는 프로그램을 작성하시오. (/\*구현\*/ 부분을 채울 것)

|  |
| --- |
| #include<iostream>  #include<iomanip>  using namespace std;  int main() {  int column = 1, row = 1;  while (/\* 구현 \*/) {  while (/\* 구현 \*/) {  int num;  /\* 구현 \*/  cout << setw(4) << num;  /\* 구현 \*/  }  cout << endl;  /\* 구현 \*/  }  return 0;  } |
|  |
| #include<iostream>  #include<iomanip>  using namespace std;  int main() {  int column = 1, row = 1;  while (row <= 10) {  column = 1;  while (column <= 10) {  int num;  if (row == column)  num = 1;  else  num = 0;  cout << setw(4) << num;  column++;  }  cout << endl;  row++;  }  return 0;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

4. 10이하의 자연수를 입력 받았을 때 성공을 출력하는 프로그램을 작성하시오. (/\*구현\*/ 부분을 채울 것)

|  |
| --- |
| 4-1. while문 사용  #include<iostream>  using namespace std;  int main() {  int num;  while (true) {  cout << "Enter the number (0, 10]: ";  cin >> num;  if (/\* 구현 \*/)  /\* 구현 \*/  }  cout << "Success!" << endl;  return 0;  } |
| 4-2. do-while 사용  #include<iostream>  using namespace std;  int main() {  int num;  do {  /\* 구현 \*/  } while (/\* 구현 \*/); //반드시 세미콜론 찍어줌  cout << "Success!" << endl;  return 0;  } |
|  |
| 4-1. while문 사용  #include<iostream>  using namespace std;  int main() {  int num;  while (true) {  cout << "Enter the number (0, 10]: ";  cin >> num;  if (num > 0 && num <= 10)  break;  }  cout << "Success!" << endl;  return 0;  } |
| 4-2. do-while 사용  #include<iostream>  using namespace std;  int main() {  int num;  do {  cout << "Enter the number (0, 10]: ";  cin >> num;  } while (num <= 0 || num > 10); //반드시 세미콜론 찍어줌  cout << "Success!" << endl;  return 0;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

5. 아래의 프로그램을 작성하시오. (/\*구현\*/ 부분을 채울 것)

|  |
| --- |
| #include<iostream>  #include<iomanip>  using namespace std;  int main() {  int n, m;  cout << "Enter n for n x m matrix: ";  cin >> n;  cout << "Enter m for n x m matrix: ";  cin >> m;  /\* 구현 \*/  return 0;  } |
|  |
| 5-1. for문 사용  #include<iostream>  #include<iomanip>  using namespace std;  int main() {  int n, m;  cout << "Enter n for n x m matrix: ";  cin >> n;  cout << "Enter m for n x m matrix: ";  cin >> m;  for (int row = 1; row <= n; row++) {  for (int col = 1; col <= m; col++) {  cout << setw(4) << row \* col;  }  cout << endl;  }  return 0;  } |
| 5-2. while문 사용  #include<iostream>  #include<iomanip>  using namespace std;  int main() {  int n, m;  cout << "Enter n for n x m matrix: ";  cin >> n;  cout << "Enter m for n x m matrix: ";  cin >> m;  int row = 1, col = 1;  while (row <= n) {  col = 1;  while (col <= m) {  cout << setw(4) << row \* col;  col++;  }  cout << endl;  row++;  }  return 0;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

6. 아래의 프로그램을 작성하시오. (/\*구현\*/ 부분을 채울 것)

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  char key;  cout << "Enter a key (p or q): ";  cin >> key;  switch (key) {  /\* 구현 \*/  default:  /\* 구현 \*/  }  return 0;  } |
|  |
| #include <iostream>  using namespace std;  int main() {  char key;  cout << "Enter a key (p or q): ";  cin >> key;  switch (key) {  case 'p':  case 'P':  cout << "Pause";  break;  case 'q':  case 'Q':  cout << "Quit";  break;  default:  cout << "Default";  break;  }  return 0;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**텍스트이(가) 표시된 사진

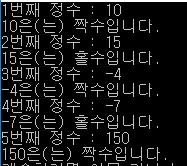
자동 생성된 설명**

**<응용문제>\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. 정수 하나를 입력 받고, 해당 정수가 짝수인지, 홀수인지를 판별하는 프로그램을 작성하시오.

* 사용자가 정수를 입력하면, 결과를 출력한 뒤에 다시 사용자에게 정수를 입력 받음.
* 해당 과정을 다섯 번 반복하고 프로그램을 종료함.

1-출력화면:



|  |
| --- |
| 응용1-1. for문 사용  #include <iostream>  using namespace std;  int main() {  int num;  for (int count = 1; count <= 5; count++) {  cout << count << "번째 정수 : ";  cin >> num;  if (num % 2 == 0)  cout << num << "은(는) 짝수입니다." << endl;  else  cout << num << "은(는) 홀수입니다." << endl;  }  return 0;  } |
| 응용1-2. while문 사용  #include <iostream>  using namespace std;  int main() {  int num;  int count = 1;  while (count <= 5) {  cout << count << "번째 정수 : ";  cin >> num;  if (num % 2 == 0)  cout << num << "은(는) 짝수입니다." << endl;  else  cout << num << "은(는) 홀수입니다." << endl;  count++;  }  return 0;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

2. 정수 n을 한 개 입력 받고, 1부터 n까지의 합을 출력하는 프로그램을 반복문을 활용하여 작성하시오.(단, n >= 1)

|  |
| --- |
| **입출력 예시:**  (input) 10  (output) 55 |

2-출력화면:



|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  int n;  int result = 0;  cout << "number : ";  cin >> n;  for (int num = 0; num <= n; num++) {  result += num;  }  cout << "Sum of 1 to " << n << " = " << result;  return 0;  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

3. 정수 n을 입력 받고, 1부터 n까지의 정수 중에서 2의 배수가 아니면서 3의 배수가 아닌 수의 총 합을 구하는 프로그램을 작성하시오. (단, n >= 1)

3-출력화면:



|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  int n;  int result = 0;  cout << "number : ";  cin >> n;  for (int num = 0; num <= n; num++) {  if (num % 2 != 0 && num % 3 != 0)  result += num;  }  cout << "Sum : " << result;  return 0;  } |

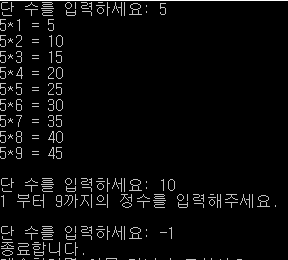
텍스트이(가) 표시된 사진

자동 생성된 설명

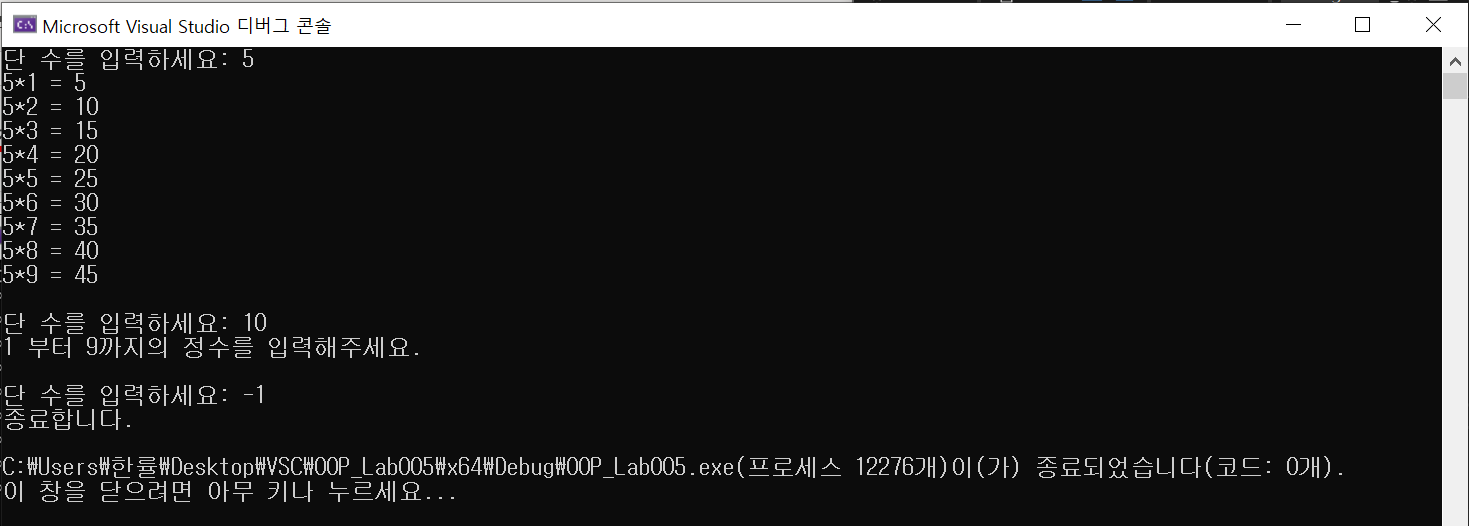
4. 사용자로부터 구구단의 ‘단’에 해당하는 숫자를 입력 받아, 해당 ‘단’을 출력하는 프로그램을 작성하시오.

* 사용자가 1~9가 아닌 숫자를 입력하면, 정상 입력 범위를 화면에 출력하고, 다시 사용자에게 ‘단’을 입력 받음.
* 사용자가 1~9 사이의 숫자를 입력하면, 해당 숫자의 구구단을 출력하고, 다시 사용자에게 ‘단’을 입력 받음.
* 사용자가 -1을 입력한 경우 프로그램을 종료함.

4-출력화면:



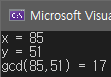
|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  int n;  while (true) {  cout << "단 수를 입력하세요: ";  cin >> n;  if (n >= 1 && n <= 9) {  for (int num = 1; num <= 9; num++) {  cout << n << "\*" << num << " = " << n \* num << endl;  }  cout << endl;  }  else if (n == -1) {  cout << "종료합니다.";  cout << endl;  break;  }  else {  cout << "1 부터 9까지의 정수를 입력해주세요." << endl;  cout << endl;  }  }  return 0;  } |



5. 두 개의 정수를 입력 받고, 두 정수의 최대공약수를 출력하는 프로그램을 재귀함수를 이용하여 작성하시오.

|  |
| --- |
| **참고 : 유클리드 호제법**  두 정수 a,b의 최대공약수를 G(a,b)라고 하자.  정수 a,b,q,r (b는 0이 아니다.)에 대하여 a = bq+r 이면 G(a,b) = G(b,r)가 성립한다.  예) G(85,51)  85 = 51\*1 + 34이기 때문에, G(85,51) = G(51,34) 이다.  G(85,51) = G(51,34) = G(34,17) = 17 즉 85와 51의 최대공약수는 17이다. |

5-출력화면:



|  |
| --- |
| #include <iostream>  using namespace std;  int gcd(int a, int b) {  if (b == 0)  return a;  else  return gcd(b, a % b);  }  int main() {  int x, y;  cout << "x = ";  cin >> x;  cout << "y = ";  cin >> y;  cout << "gcd(" << x << "," << y << ") = " << gcd(x, y) << endl;  return 0;  } |

