Name: Hemos ID:

Student ID (학번):

CSED-101 PROGRAMMING & PROBLEM SOLVING

Spring 2014

Midterm

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Q11 | Q12 | Total |
| Your Score |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Max  Score | 5 | 18 | 7 | 4 | 6 | 5 | 7 | 7 | 10 | 8 | 16 | 7 | 100 |

* Write down your name, hemos ID, and student ID.
* There are 14 pages in this midterm.
* Your answers must run correctly in **C programming language** without error or warning. Otherwise, your answers will be considered incorrect. For example, it is ok to put more parentheses than needed in your answer, but it will be incorrect if you put fewer parentheses than needed.
* You must write your answer on the underline => \_\_\_\_\_\_\_\_. Scratches outside the underline will be ignored.
* The total score is 100
* This is a 3-hour exam.

|  |
| --- |
| **명예서약서**  **나는 명예로운 포스테키안으로서 다음의 Honor Code를 지킬 것을 맹세합니다.**  **정직과 타인에 대한 존중이 함께하는 포스테키안의 미래는 명예롭다.**  **2014년 월 일**  **이름 (인)** |

1. (5점) 아래 condition의 결과가 true 이면 T, false 이면 F 로 답하시오.

단, 각 문항 별 1점 배점, 오답일 경우 -1점, 공백의 경우 0점.

int a = 1, b = 2, c = 3, d = 4, e = 5, x = -9, y = -3, z = 0;

* 1. [ \_\_\_\_\_\_F\_\_\_\_ ] a < c > b
  2. [ \_\_\_\_\_\_T\_\_\_\_ ] 4 \* d / 5 % 3 && d
  3. [ \_\_\_\_\_\_F\_\_\_\_ ] d == (e++) – 1
  4. [ \_\_\_\_\_\_T\_\_\_\_ ] y == (a += b\*= c -=5)
  5. [ \_\_\_\_\_\_F\_\_\_\_ ] z || x < (y – 5) && !y

1. (18점) 다음 문제를 읽고 답하시오.
   1. (1점) 다음 중 변수명으로 사용할 수 없는 것은? \_\_\_\_\_2\_\_\_\_\_

① cs101 ②$option ③ A4 ④ IF ⑤ \_2dNum

* 1. (1점) 다음 중 함수 선언(declaration)이 올바르게 된 것은? \_\_\_3\_\_\_\_\_\_\_

① int fun(int &x, int y);

② int fun(int x, y);

③ void fun(float x, int y);

④ void fun(x, y);

* 1. (3점) x=1, y=2, z=3 인 경우, 다음 코드를 실행 시킨 후의 x, y, z 의 값은?

|  |  |
| --- | --- |
| switch ( z > y )  {  case 0: x--;  y--;  case 1: x--;  z--;  default: y--;  z--;  } | x : \_\_\_\_\_\_\_\_\_\_  y : \_\_\_\_\_\_\_\_\_\_  z : ­­\_\_\_\_\_\_\_\_\_\_ |

* 1. (2점) x=10, y=40 인 경우, 다음 코드를 실행 시킨 후의 x 의 값은?

|  |  |
| --- | --- |
| do  {  x += 9;  }while(x < y); | x : \_\_\_46\_\_\_\_\_\_\_ |

* 1. (3점) x=0, y=1 인 경우, 다음 코드를 실행 시킨 후의 x, y 의 값은?

|  |  |
| --- | --- |
| for(x = 10; x > 0; x = x – 3)  {  y = y \* x;  } | x : \_\_\_\_\_\_\_\_\_\_  y : \_\_\_\_\_\_\_\_\_\_ |

* 1. (2점) x=0, y=0 인 경우, 다음 코드를 실행 시킨 후의 x, y 의 값은?

|  |  |
| --- | --- |
| for(x = 0; x < 10; x++)  {  if ( x % 2 == 0 )  continue;  y += x;  } | x : \_\_\_\_\_\_\_\_\_\_  y : \_\_\_\_\_\_\_\_\_\_ |

* 1. (4점) x=0, y=0, z=0 인 경우, 다음 코드를 실행 시킨 후의 x, y, z 의 값은?

|  |  |
| --- | --- |
| for (x = 0; x < 10; x += 3)  for (y = x + 1; y < 10; y += 5)  z += x + y; | x : \_\_\_\_\_\_\_\_\_\_  y : \_\_\_\_\_\_\_\_\_\_  z : \_\_\_\_\_\_\_\_\_\_ |

* 1. (2점) x=10, y=10, z=0 인 경우, 다음 코드를 실행 시킨 후의 z 의 값은?

|  |  |
| --- | --- |
| while (x--)  do  z = x + y;  while (--x); | z : \_\_\_\_\_\_\_\_\_\_ |

1. (7점) 아래의 프로그램을 실행하였을 때, 화면에 출력되는 결과를 쓰시오.

|  |
| --- |
| #include <stdio.h>  #define ADD 6 + 3  int f1(int a, int b);  int f2(int a, int b, int c);  int f3(int a, int b);  void swap(int a, int b);  int main(void)  { int a = 2, b = 4, c = 5;  printf(“%d %d\n”, a, f1(a, b)); //(1)  c += f2(a, b, c);  printf(“%d\n”, c); //(2)  printf(“%d\n”, f3(c, 0)); //(3)  swap(a, b); //(4)  printf(“%d %d\n”, a, b); //(5)  printf(“%d\n”, ADD / 3); //(6)  return 0;  }  int f1(int a, int b)  {  b = a + b;  a = 10;  return b;  }  int f2(int a, int b, int c)  {  --a; --b; --c;  return –a;  }  int f3(int a, int b)  {  for (a = -4; a < 4; a++)  if (a + 1) b += a;  return b;  }  void swap(int a, int b)  {  int temp = a;  a = b;  b = temp;  printf(“%d %d\n”, a, b);  } |

출력 결과

|  |  |
| --- | --- |
| (1) (1점) |  |
| (2) (1점) |  |
| (3) (2점) |  |
| (4) (1점) |  |
| (5) (1점) |  |
| (6) (1점) |  |

1. (4점) 아래 프로그램은 사용자로부터 두 숫자와 하나의 문자를 순서대로 입력 받아, 사칙 연산을 계산하는 프로그램이다. 이 프로그램이 정상적으로 작동할 수 있도록 빈 칸을 채우시오.

|  |
| --- |
| #include <stdio.h>  int main(void)  {  float a, b;  char c;  while (1)  {  scanf(“%f %f”, &a, &b);  scanf(“ %c”, &c);  if (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)  {  printf(“%f + %f = %f\n”,\_\_\_\_\_,\_\_\_\_\_,\_\_\_\_\_);  }  else if (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)  {  printf(“%f - %f = %f\n”,\_\_\_\_\_,\_\_\_\_\_,\_\_\_\_\_);  }  else if (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)  {  printf(“%f \* %f = %f\n”,\_\_\_\_\_,\_\_\_\_\_,\_\_\_\_\_);  }  else if (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)  {  printf(“%f / %f = %f\n”,\_\_\_\_\_,\_\_\_\_\_,\_\_\_\_\_);  }  else  {  printf(“End\n”);  break;  }  }  return 0;  } |

1. (6점) 아래의 프로그램은 4개의 정수를 입력 받아, 덧셈 및 뺄셈을 수행한 후 그 결과를 출력하는 프로그램이다. 아래의 실행 결과를 참조하여 빈 칸을 채우시오. 단, +, -, \*, /, %와 같은 연산자를 사용하지 않고, 오직 **add** 와 **sub** 함수만을 이용해서 작성한다.

**(실행예시)**

|  |
| --- |
| Insert four integer values: 1 2 3 4  1+2+3+4=10  1-2-3-4=-8 |

|  |
| --- |
| #include <stdio.h>  int add(int x, int y);  int sub(int x, int y);  int main()  {  int a, b, c, d, e, f;  printf(“Insert four integer values: “);  scanf(“%d %d %d %d”, &a, &b, &c, &d);    e = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;    f = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  printf(“%d+%d+%d+%d=%d\n”, a, b, c, d, e);  printf(“%d-%d-%d-%d=%d\n”, a, b, c, d, f);  return 0;  }  int add(int x, int y)  {  return x + y;  }  int sub(int x, int y)  {  return x – y;  } |

1. (5점) 아래 프로그램은 주사위 2개를 굴려서 나온 숫자의 합이 짝수면 사용자가 이기고, 홀수면 지는 프로그램이다. 주사위에서 나올 수 있는 숫자의 범위는 1~6이다. 프로그램이 정상 작동하도록 빈 칸을 채우시오.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <time.h>  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  void printResult(int a);  int main(void)  {  int dice1, dice2;  srand(time(NULL));  dice1 = randDice();  dice2 = randDice();  printf(“Dice 1: %d, Dice 2: %d\n”, dice1, dice2);  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  return 0;  }  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  {  return \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  }  void printResult(int a)  {  if (\_\_\_\_\_)  printf(“The sum of the 2 dice is %d. You lose.\n”, a);  else  printf(“The sum of the 2 dice is %d. You win.\n”, a);  } |

1. (7점) 아래 프로그램은 3자리 양의 정수를 랜덤하게 생성하려는 프로그램이다. 단, 각 자리의 숫자는 서로 중복되는 경우가 없다는 조건을 만족시켜야 한다. 빈 칸을 채우시오.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <time.h>  #define TRUE 1  #define FALSE 0  int is\_duplicate(int n);  int main()  {  int num;  srand(time(NULL));  do  {  num = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  }while(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_);  printf(“%d\n”, num);  return 0;  }  int is\_duplicate(int n)  {  int n100, n10, n1;  int check = FALSE;  n1 = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  n10 = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  n100 = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  if(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)  check = TRUE;  return \_\_\_\_\_\_\_\_\_;  } |

1. (7점) 피보나치 수는 아래의 점화식으로 정의되는 수열이다.
   1. (4점) 피보나치 수를 ***int fibo (int n)*** 이라는 함수를 사용하여 구현하고자 한다. 이 함수를 **recursion**을 이용하여 작성하시오.

|  |
| --- |
| #include <stdio.h>  int fibo(int n);  int main(void)  {  int seriesSize = 12;  int looper;  printf(“First %d Fibonacci numbers:”, seriesSize);  for (looper = 0; looper < seriesSize; looper++)  {  if(looper % 4)  printf(“, %4d”, fibo(looper));  else  printf(“\n%4d”, fibo(looper));  }  printf(“\n”);  return 0;  }  int fibo(int n)  {  if (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)  return \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  else  return \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  } |

* 1. (3점) 위의 작성된 프로그램을 실행하였을 때, 화면에 출력되는 결과를 쓰시오.

|  |
| --- |
|  |

1. (10점) 다음 문제를 읽고 답하시오.
   1. (5점) 아래 프로그램은 사용자로부터 양의 정수 1개를 입력 받아, 숫자의 순서를 거꾸로 출력하는 프로그램이다. 프로그램이 정상 작동하도록 빈 칸을 채우시오.

|  |
| --- |
| #include <stdio.h>  void reverseInt(int n);  int main(void) {  int n;  scanf(“%d”, &n);  reverseInt(n);  return 0;  }  // 아래의 함수는 recursion 을 이용하여 완성하시오. (재귀 함수)  void reverseInt(int n)  {  if(\_\_\_\_\_\_\_\_\_) // (1점)  printf(“%d”, n);  else  {  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  }  } |
| **[실행 결과]**  **입력: 54321**  **출력: 12345** |

* 1. (5점) 위의 함수 **reverseInt를 while문**을 이용하여 완성하시오. (재귀 함수로 작성할 수 없음)

|  |
| --- |
| void reverseInt(int n)  {  while(\_\_\_\_\_\_\_\_\_\_)  {  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  }  } |

1. (8점) 다음은 주어진 배열의 순서를 뒤집는 프로그램이다. 배열의 각 원소는 처음에는 배열에서의 자신의 위치 값을 할당한다. 프로그램이 정상적으로 작동하도록 빈칸을 채우시오. 추가적인 변수는 선언할 수 없으며 ARY\_SIZE를 최소 6번 사용해야 한다.

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #define ARY\_SIZE 10  int main()  {  \_\_\_\_\_\_\_\_\_\_\_\_\_\_  int i;  int temp;  // 초기 값을 지정해준다.  for (i = 0; i < \_\_\_\_\_\_\_\_\_\_\_\_ ; i++)  {  \_\_\_\_\_\_\_\_\_\_\_\_\_\_  printf(“%d “, \_\_\_\_\_\_\_\_\_\_\_);  }  printf(“\n”);  // 배열의 순서를 뒤집어 저장한다.  for (i = 0; i < \_\_\_\_\_\_\_\_\_\_\_\_ ; i++)  {  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  }  // 바뀐 값을 출력한다.  for (i = 0; i < \_\_\_\_\_\_\_\_ ; i++)  {  printf(“%d “, \_\_\_\_\_\_\_);  }  printf(“\n”);  return 0;  } |
| **실행결과:**  0 1 2 3 4 5 6 7 8 9  9 8 7 6 5 4 3 2 1 0 |

1. (16점) 아래의 프로그램은 파일(input.txt)로부터 학생들의 시험 성적을 읽고 각 학생의 총점(평균으로 계산)과 학점 및 과목평균을 계산하여 화면에 출력하는 프로그램이다. 아래 빈칸을 채워 프로그램을 완성하시오. 단, 2개의 변수만 선언해서 사용할 수 있다.

파일에는 줄(line) 단위로 한 학생의 학번, 중간고사 점수, 기말고사 점수가 순서대로 기록되어 있다.

**(input.txt)** **(출력화면)**

|  |
| --- |
| 201301 89 78  201302 76 84  201303 67 42  201304 66 88  201305 98 99 |

|  |
| --- |
| **================**  **학번 총점(학점)**  **================**  **201301 83.5 (B)**  **201302 80.0 (B)**  **201303 54.5 (F)**  **201304 77.0 (C)**  **201305 98.5 (A)**  **================**  **과목평균: 78.70**  **================** |

|  |
| --- |
| #include <stdio.h>  float calc\_average(int x, int y);  char calc\_grade(float score);  int main()  {  FILE \*infile;  int stu\_id, mid, final;  float average;  char grade;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  infile = fopen(\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_);  if (infile \_\_\_\_\_\_\_\_\_\_\_)  {  printf(“cannot open input file.\n”);  return 1;  }  printf(“=================\n”);  printf(“학번\t총점(학점)\n”);  printf(“=================\n”);  while( fscanf(\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) \_\_\_\_\_\_)  { // (5점) 아래의 5줄 중 4줄만 채우면 됨. 아래에서 변수 선언 할 수 없음  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  printf(“%d\t%.1f (%c)\n”, stu\_id, average, grade);  }  printf(“=================\n”);  printf(“과목평균: %.2f\n”, tot\_ave / num\_of\_stu );  printf(“=================\n”);  fclose(infile);  return 0;  } // main  // 함수의 입력 값으로 전달 받은 x와 y의 평균을 계산  float calc\_average(int x, int y)  {  return \_\_\_\_\_\_\_\_\_\_\_\_  }  // 학점 계산  char calc\_grade(float score)  {  char grade;  switch ( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ )  {  case 10:  case 9 : grade = ‘A’;  break;  case 8 : grade = ‘B’;  break;  case 7 : grade = ‘C’;  break;  case 6 : grade = ‘D’;  break;  default: grade = ‘F’;  } //switch  return grade;  } |

1. (7점) 아래의 프로그램은 배열의 원소 중 가장 큰 값을 찾아 그 값을 출력하는 프로그램이다. 빈 칸을 채우시오.

배열 list 는 아래의 그램과 같이 정수형의 원소로 구성되어 있다.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
| list | 0 | 5 | 7 | 3 | 2 | 6 | 9 | 1 | 4 | 8 |

|  |
| --- |
| #include <stdio.h>  #define ARY\_SIZE 10  int get\_max(int arr[], int size);  int main()  {  // (2점) 배열 list 를 위의 그림과 같이 초기화 하시오.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  int max\_value;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ // (2점)함수 get\_max() 호출  printf(“MAX = %d\n”, max\_value);  return 0;  } // main  int get\_max(int arr[], int size)  {  int i;  int max = \_\_\_\_\_\_\_\_\_\_\_\_\_\_;  for (i = 1; i < size; i++)  {  if( \_\_\_\_\_\_\_\_\_\_\_\_\_\_ )  {  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  }  } //for  return max;  } // get\_max |
| **실행 결과:**  MAX = 9 |