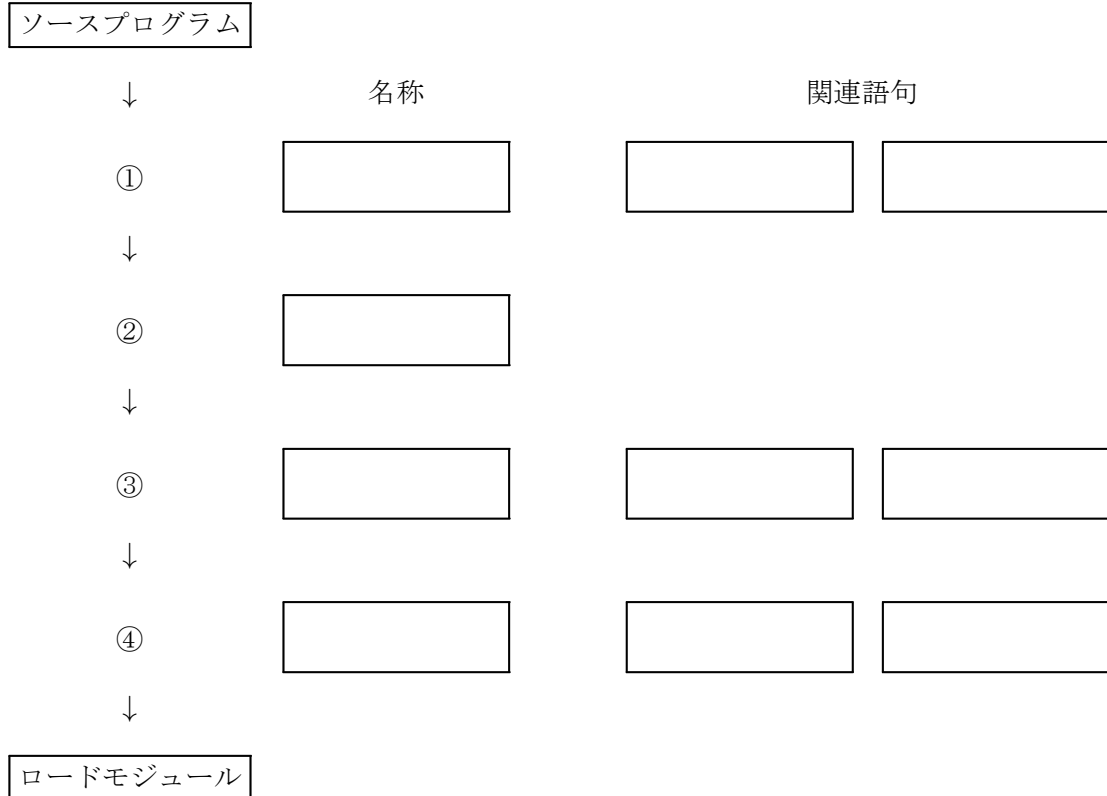


2009 年度 情報科学&情報科学演習 期末試験 (その 1)

学籍番号： _____

氏名： _____

問題 1 以下の図は、C 言語によるプログラミングにおいて、ソースプログラムからロードモジュール (実行可能プログラム) を生成するまでの過程を示したものである。図の各過程の名称および関連語句について選択肢から適切な語句を選び、A～L の記号で答えなさい。(10 点)



選択肢：

- | | | | |
|------------|------------|----------|----------------|
| A. トレース | B. プリプロセス | C. less | D. オブジェクトモジュール |
| E. ヘッドファイル | F. 機械語 | G. リンク | H. 標準ライブラリ |
| I. アセンブル | J. #define | K. コンパイル | L. リンカ |

2009年度 情報科学&情報科学演習 期末試験 (その2)

学籍番号：

氏名：

問題 2 次のプログラム「配列の内容を表示するプログラム」を作成したが、コンパイルするとエラーが生じた。この後、プログラムを5箇所修正したところ正しくコンパイルされ、正しい実行結果を得ることができた。修正すべき箇所の行番号と修正文を解答欄に記入しなさい。(15点)

● 配列の内容を表示するプログラム

printarray.c

```
1: #include <stdlib.h>
2:
3: void printarray(double a[], int n)
4:
5: int main(void)
6: {
7:     double a[] = {1.1, 3.56, -5.45, -4.03, 2.0752};
8:
9:     printarray(a);
10:
11:     return 0;
12: }
13:
14: void printarray(double a[], int n)
15: {
16:     int i;
17:
18:     for (i = 0, i < n, i++) {
19:         printf("%d ", a[i]);
20:     }
21: }
```

解答欄：

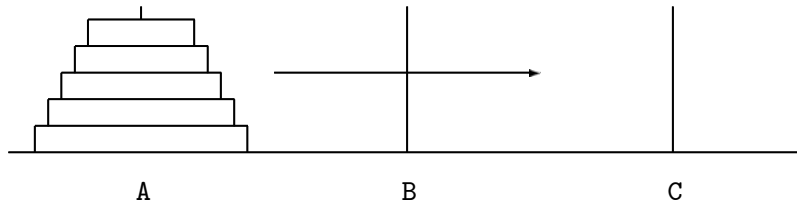
行番号	修正文

2009 年度 情報科学&情報科学演習 期末試験 (その 3)

学籍番号： _____

氏名： _____

問題 3 次のプログラム「ハノイの塔 (再帰的)」は、ハノイの塔の最短手順を再帰的に求めるプログラムである (下図のように棒「A」から棒「C」に大小互いに異なる 5 つの円盤を移す)。次の (1)~(2) の問いに答えなさい。



ハノイの塔のルール：

- 一度に 1 枚の円盤しか動かしてはならない。
- 小さい円盤の上に大きい円盤を重ねてはならない。

● ハノイの塔 (再帰的)

hanoi.c

```
1: #include <stdio.h>
2:
3: void hanoi(int n, char x, char y, char z);
4:
5: int main(void)
6: {
7:     int n = 5;
8:
9:     /* printf("円盤の枚数?: "); */
10:    /* scanf("%d", &n); */
11:    hanoi(n, 'A', 'B', 'C');
12:
13:    return 0;
14: }
15:
16: void hanoi(int n, char x, char y, char z)
17: {
18:     if (n > 1) hanoi(n - 1, x, z, y);
19:     printf("円盤 %d を %c から %c に移す\n", n, x, z);
20:     if (n > 1) hanoi(n - 1, y, x, z);
21: }
```


2009 年度 情報科学&情報科学演習 期末試験 (その 5)

学籍番号: _____

氏名: _____

問題 4 次のプログラム「ヒープソートによるソーティングプログラム (昇順)」を実行したとき、第 33 行 (ヒープ化) と第 51 行 (ダウンヒープ) における配列「a[]」の状態 (関数「printarray()」による出力結果) を解答欄に記入しなさい。(10 点)

● ヒープソートによるソーティングプログラム (昇順)

heapsort.c

```
1: #include <stdio.h>
2:
3: #define N 10
4:
5: void heapsort(int a[]);
6: void printarray(int a[]);
7:
8: int main(void)
9: {
10:     int a[N+1] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
11:
12:     heapsort(a);
13:
14:     return 0;
15: }
16:
17: void heapsort(int a[])
18: {
19:     int i, j, k, n = N, x;
20:
21:     for (k = n/2; k >= 1; k--) {
22:         i = k;
23:         x = a[i];
24:         j = 2 * i;
25:         while (j <= n) {
26:             if (j < n && a[j] < a[j+1]) j++;
27:             if (x >= a[j]) break;
28:             a[i] = a[j];
29:             i = j;
30:             j = 2 * i;
31:         }
32:         a[i] = x;
33:         printarray(a);
34:     }
35:     printf("%n");
```

2009 年度 情報科学&情報科学演習 期末試験 (その6)

学籍番号: _____

氏名: _____

● ヒープソートによるソーティングプログラム (昇順)	... 続き ...	heapsort.c
-----------------------------	------------	------------

```
36:
37:     while (n > 1) {
38:         x = a[n];
39:         a[n] = a[1];
40:         n--;
41:         i = 1;
42:         j = 2 * i;
43:         while (j <= n) {
44:             if (j < n && a[j] < a[j+1]) j++;
45:             if (x >= a[j]) break;
46:             a[i] = a[j];
47:             i = j;
48:             j = 2 * i;
49:         }
50:         a[i] = x;
51:         printarray(a);
52:     }
53: }
54:
55: void printarray(int a[])
56: {
57:     int i;
58:
59:     for (i = 1; i <= N; i++) printf("%3d ", a[i]);
60:     printf("\n");
61: }
```

【注意】

- このプログラムにおけるソーティングの範囲は「a[1]」～「a[10]」である。したがって、このプログラムによって配列「a[0]=0」は影響を受けない。
- このプログラムにおいて、配列を2分木に対応させたとき、親「a[i]」に対する2つの子は「a[2 * i]」と「a[2 * i + 1]」である。

2009 年度 情報科学&情報科学演習 期末試験 (その 7)

学籍番号： _____

氏名： _____

問題 4 の解答欄：

初期状態： * 関数「`printarray()`」では出力されない。

{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }

ヒープ化： * 第 33 行における配列「`a[]`」の状態

{ 0, , , , , , , , , , 5 }
{ 0, , , , , , , , , , }
{ 0, , , , , , , , , , }
{ 0, , , , , , , , , , }
{ 0, , , , , , , , 1, , }

ダウンヒープ： * 第 51 行における配列「`a[]`」の状態

{ 0, , , , , , , , , , }
{ 0, , 5, , , , , , , , }
{ 0, , , , , , 1, , , , }
{ 0, , , , , , , , , , }
{ 0, , , , , , , , , , }
{ 0, 4, 2, 3, 1, 5, 6, 7, 8, 9, 10 }
{ 0, 3, 2, 1, 4, 5, 6, 7, 8, 9, 10 }
{ 0, 2, 1, 3, 4, 5, 6, 7, 8, 9, 10 }
{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }