平成20年度大学院前期課程入試試験問題

選択科目　b. システム制御

平成19年8月21日

注意事項

- 問題用紙は全部で9 枚（但し、表紙を除く）あるので確認すること。
- 解答には必ず問題番号を書き、どの問題に解答したかわかるようにすること。
- 「制御工学」（問題1および2）は全員が解答せよ。
- 選択問題（問題3～7）から2分野を選択して解答せよ。選択しなかった問題の解答用紙には×印を記して選択した問題が明確に分かるようにせよ。
- 選択問題から3分野以上解答した場合には選択問題の解答を全て無効とするので注意せよ。
- 「電気機器」を選択する者は、問題3－1および3－2を解答せよ。
- 「パワーエレクトロニクス」を選択する者は、問題4を解答せよ。
- 「信号処理」を選択する者は、問題5を解答せよ。
- 「論理回路・計算機システム」を選択する者は、問題6を解答せよ。
- 「基本アルゴリズム・プログラミング」を選択する者は、問題7－1および7－2を解答せよ。
- 解答用紙は色分けしてあるので、問題番号と対応させて以下のように使い分けよ。（間違わないように注意せよ。）

問題番号	解答用紙の色
1	白
2	赤
3	青（紺）
4	黄
5	水（薄い青）
6	桃
7	緑

- 解答用紙の表に書き切れない場合は裏を使用しても良い。
- 問題用紙は持ち帰っても良い。

制御工学

1．図１－１のフィードバック制御系について以下の問いに答えよ．ただし，$G(s) = \dfrac{12}{s^3+9s^2+20s}$，$K>0$ とする．

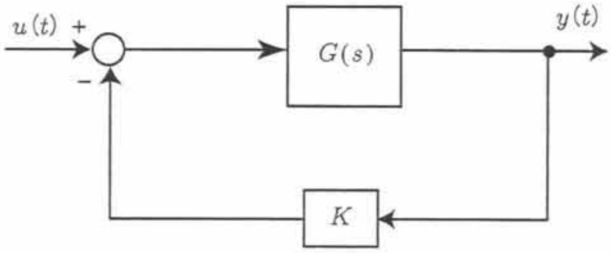


図１－１

(i) $G(s)$ が図１－２で構成されている場合，K_A, K_T, a の値を求めよ．

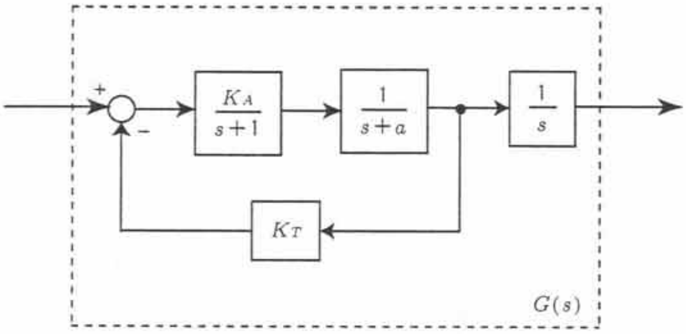


図１－２

(ii) $G(s)$ のボード線図（ゲイン特性ならびに位相特性）の概形を描け．なお，ゲイン特性は，折点周波数を明示して折線近似で示せ．また，位相特性は，$\omega \to 0$, $\omega \to \infty$ のときの漸近値ならびに位相が $-180°$ となる角周波数を明示して描け．

(iii) $G(s)$ のベクトル軌跡の概形を示せ．特に $\omega \to 0$, $\omega \to \infty$ のときの形状に注意して描け．

(iv) 図１－１の閉ループ系のゲイン余裕が 20dB となるような K の値を求めよ．

(v) 図１－１の閉ループ系が安定となる K の条件を求めよ．

(vi) 図１－１の閉ループ系が特性根の１つに -8 を持つとき，K ならびに支配極を求めよ．

(vii) $K=1$ の場合の図１－１の閉ループ系における単位ステップ応答 $y(t)$ を求めよ．

2．図2は，重力で落下しようとする鉄球（質量 M の質点とみなす）を電磁石により保持するための磁気浮上システムである．鉄球は鉛直方向にのみ移動可能とし，電磁石端を原点とする鉄球の時刻 t における位置を $x(t)$ （図で矢印の向きが正），電磁石に流れる電流を $i(t)$ とすると，鉄球が電磁石から受ける吸引力は，$k(i(t)/x(t))^2$ で与えられるので（k は電磁石の吸引力係数），このシステムの運動方程式は

$$M\frac{d^2x(t)}{dt^2} = Mg - k\left(\frac{i(t)}{x(t)}\right)^2$$

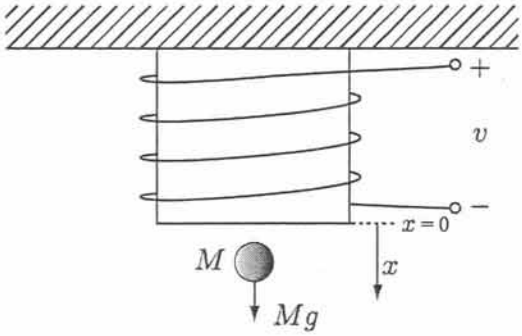


図2　磁気浮上システム

となる．ここで，g は重力加速度であり，$M, g, k > 0$ である．電磁石の巻線のインダクタンスを $L > 0$，抵抗を $R > 0$ とすると，電磁石に加える電圧 $v(t)$ と電流 $i(t)$ の間には

$$L\frac{di(t)}{dt} + Ri(t) = v(t)$$

が成り立つ（ここで，インダクタンス L は鉄球の位置に依存せず定数であると仮定した）．電圧 $v(t)$ をうまく調節することにより，鉄球を電磁石に非接触のまま，安定に保持したい．以下の問いに答えよ．

(i) 新たな変数を，$x_1(t) = x(t)$，$x_2(t) = dx(t)/dt$，$x_3(t) = i(t)$ と導入する．変数 x_1, x_2, x_3 に関する状態方程式（1階連立非線形微分方程式）を求めよ．

(ii) 電磁石の吸引力と重力が釣り合っているときの，位置 x，電流 i，電圧 v の値をそれぞれ x^*, i^*, v^* とする．i^* と x^* との間に成り立つ関係および i^* と v^* との関係を求めよ．

(iii) 簡単のため，以下では $L = 1$ とする．前問における釣り合いの状態からの微小変位を $z_1(t) = x_1(t) - x^*$，$z_2(t) = x_2(t)$，$z_3(t) = x_3(t) - i^*$，$u(t) = v(t) - v^*$ と表し，問い (i) で求めた状態方程式を線形化すると，z_1, z_2, z_3 に関する状態方程式は

$$\frac{d}{dt}\begin{pmatrix} z_1(t) \\ z_2(t) \\ z_3(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ \alpha & 0 & -\beta \\ 0 & 0 & -\gamma \end{pmatrix}\begin{pmatrix} z_1(t) \\ z_2(t) \\ z_3(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u(t), \qquad \alpha, \beta, \gamma > 0 \tag{1}$$

となった．α, β, γ を i^*, x^*, M, k, R を用いて表せ．

(iv) (1) 式のシステムにおいて，入力 u から状態変数 z_1 への伝達関数 $G(s)$ を α, β, γ を用いて表せ．また，このシステムが安定かどうか，理由とともに述べよ．

(v) (1) 式のシステムの可制御性を判定せよ．

(vi) (1) 式のシステムへの入力として，状態フィードバック $u(t) = k_1z_1(t) + k_2z_2(t) + k_3z_3(t)$ を用いる．状態フィードバックされたシステムの特性方程式 $\phi(s) = 0$ が根 $s = -\lambda, -\sigma \pm j\omega$, $(\lambda, \sigma, \omega > 0$, j は虚数単位) を持つように極指定することは可能か？　可能な場合は，そのときの k_1, k_2, k_3 を $\alpha, \beta, \gamma, \lambda, \sigma, \omega$ を用いて表し，不可能な場合は理由を述べよ．

(vii) 前問と同じ状態フィードバックを施したシステムを考える．電流の値を用いずに（$k_3 = 0$），位置と速度だけをフィードバックすることで，システムを安定化することは可能か？　可能な場合は，安定化のための k_1, k_2 に関する条件を α, β, γ を用いて表し，不可能な場合は理由を述べよ．

電気機器

3－1　三相誘導電動機について，以下の問いに答えよ．

(i) 6 極，定格 220 V，60 Hz，1140 rpm (rpm: 毎分の回転数)の三相巻線形誘導電動機がある．1 次巻線，2 次巻線ともに三相でΔ結線されており，2 次側挿入抵抗なしの場合の 2 次巻線一相あたりの抵抗は r_2 = 0.04 [Ω]，始動時の漏れリアクタンスは x_2 = 0.6 [Ω]である．1 次側のインピーダンスは無視する．また，巻線比(巻線係数と巻数比の積)は $a = 8$ とする．次の値を求めよ．

(a) 定格運転中のすべり s

(b) 始動時の 2 次誘導電圧 E_2

(c) 定格運転中の 2 次誘導電圧 E_{2s}

(d) 定格運転中の 2 次周波数 f_{2s}

(e) 定格運転中の 2 次回路のインピーダンスの大きさ Z_{2s}

(f) 定格運転中の 2 次相電流 I_{2sp}

(g) 定格運転中の三相分の機械出力 P_{M}

(ii) この三相誘導電動機にポンプ負荷を接続した．このポンプ負荷のトルクは回転速度の 2 乗に比例して変化するものとする．一方，すべりが十分小さい領域では，誘導電動機のトルクはすべりに比例すると考えることができる．いま 2 次側挿入抵抗なしの場合で，100%負荷トルク運転時の回転速度が 1140 rpm である．2 次回路に抵抗を挿入して，912 rpm に回転速度を制御するには一相あたりいくらの抵抗を挿入したらよいか，答えよ．

3－2　直流機について以下の問いに答えよ．ただし，各抵抗値は運転中変わらないものとし，ブラシによる電圧降下，電機子反作用・飽和は無視する．

(i) 電機子抵抗 R_{a} が 0.1 Ωである直流分巻発電機が，回転数 N_{g} = 2000 [rpm]で運転しているときに，端子電圧 V=220 [V]，電機子電流（負荷電流 I ではないことに注意）I_{a}=50 [A]であった．いまこれを電動機として使用し，その端子電圧及び電機子電流を発電機運転の場合と同一にするとき，その回転数 N_{m} [rpm]を求めよ．

(ii) 図 3 に示す直流分巻電動機の電機子抵抗 R_{a} が 0.1 Ω，界磁抵抗（界磁巻線の抵抗を含む）R_{f} が 25 Ωのとき，端子電圧 V = 100 [V]で電源から I = 104 [A]を供給した場合に回転数は 1000 rpm であった．同一の端子電圧で 102 A を供給し，回転数を 2000 rpm とするには，界磁抵抗 R_{f} をいくらにしたらよいか，答えよ．

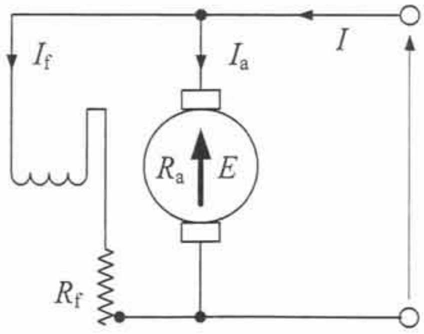
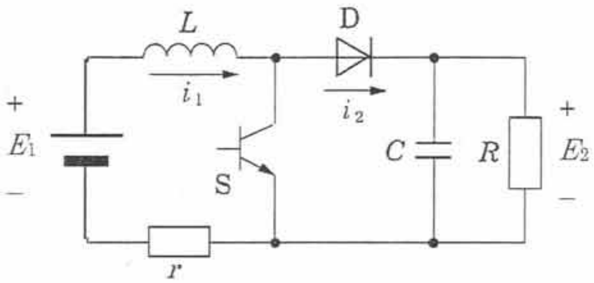


図 3

パワーエレクトロニクス

4．図に示す抵抗 R を負荷に持つチョッパ回路において，スイッチング素子 S を周期 T でオン・オフ制御を行う．すなわち，$0<t\leqq T_{on}$ で S はオン，$T_{on}<t\leqq T_{on}+T_{off}=T$ でオフとし，以後同様に周期 T でオン・オフを繰り返すものとする．このとき，以下の各問いに答えよ．ただし，キャパシタ C の静電容量はその両端電圧が一定と見なせる程度に十分に大きく，$t<0$ で定常状態の電圧値 E_2 まで充電されている．また，インダクタ L の電流は決してゼロにはならないものとし，さらにスイッチング素子 S およびダイオード D の特性は理想的であるとして，オン時の電圧降下，オフ時の漏れ電流，およびスイッチング時間は無視する．なお，抵抗 r は電源の内部抵抗を表すものとする．



（i）インダクタ L の電流変化に着目し，S がオンのときの等価回路およびオフのときの等価回路を示せ．

（ii）S がオンのときの回路の微分方程式を書き，それを解くことにより $t=T_{on}$ におけるインダクタ L の電流の式を求めよ．ただし，$t=0$ におけるインダクタ L の電流は I_{10} であるとする．

（iii）S がオフのときの回路の微分方程式を書き，それを解くことにより $t=T_{on}+T_{off}\ (=T)$ におけるインダクタ L の電流の式を求めよ．ただし，$t=T_{on}$ におけるインダクタ L の電流は I_{20} であるとする．

（iv）スイッチング動作を繰り返し，回路が定常状態になったとする．この状態でのインダクタ L の電流の波形を示せ．なお，スイッチング時の電流の境界条件の関係が良く分かるように波形を描くこと．

（v）問い（iv）で示した定常状態の波形における I_{10} および I_{20} を求めよ．

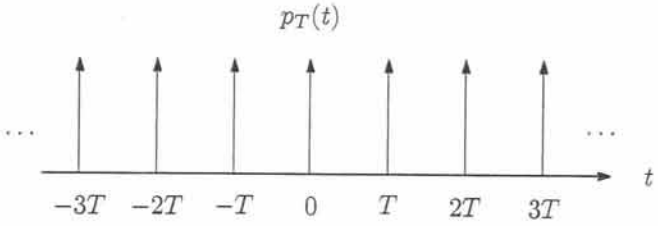
（vi）定常状態における E_1 と E_2 の関係式を求めたい．以下の（　）に適する式を示せ．なお，電流 i_1 および i_2 の変化は直線で近似できるものとする．

定常状態において，インダクタ L の両端電圧のスイッチング周期 T での平均値はゼロであることから，（ ① ）E_1 +（ ② ）E_2 +（ ③ ）$I_1=0$　((1)式) が成立する．ただし，I_1 は電流 i_1 のスイッチング周期での平均値である．次に，定常状態ではキャパシタ C の電流のスイッチング周期 T での平均値はゼロであることから電流 i_2 のスイッチング周期での平均値 I_2 =（ ④ ）E_2 となる．また，I_1 と I_2 との間には，I_1 =（ ⑤ ）I_2 の関係がある．これらの関係から(1)式における I_1 を消去すると，E_2 =（ ⑥ ）E_1 となる．

（vii）スイッチング素子 S として用いることが出来る半導体素子の名称（略称で可）を 1 種類示し，かつ，その素子の構造および動作の特徴について合計 40 字程度で述べよ．

信号処理

5．連続時間信号 $x(t)$ のサンプリングを考える．無限に続く周期 T のサンプリングパルス列信号 $p_T(t)$ （サンプリング周波数 $f_S = \dfrac{1}{T}$）を下図のようにする．以下の問いに答えよ．



(i) 単位インパルス信号（ディラックのデルタ関数 $\delta(t)$）を用いて $p_T(t)$ を表し，フーリエ級数展開せよ．

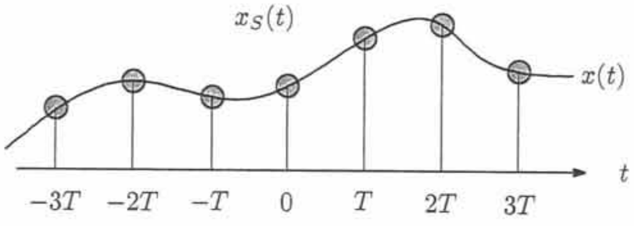
ここで，$x(t)$ のフーリエ変換を $X(\omega)$ とし，

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$

と定義する（j は虚数単位）．フーリエ変換対を $x(t) \leftrightarrow X(\omega)$ と書く．また $X(\omega)$ はスペクトルとも呼ぶ．いま，$x(t)$ と $p_T(t)$ の積をとった信号 $x_S(t)$ は

$$x_S(t) = x(t) \cdot p_T(t) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT)$$

となる（下図参照）．続けて以下の問いに答えよ．



(ii) $e^{j\omega' t} \leftrightarrow 2\pi\delta(\omega - \omega')$ を利用して $p_T(t)$ のフーリエ変換 $P_T(\omega)$ を求めよ．

(iii) $x_S(t)$ のフーリエ変換 $X_S(\omega)$ を求めよ．

(iv) $x(t)$ が帯域制限されているとき，すなわち

$$X(\omega) = 0, \quad |\omega| > \omega_M$$

のとき，問い (iii) の $X_S(\omega)$ より $X(\omega)$ が正しく抽出される条件を求めよ．

論理回路・計算機システム

6．3つの2ビット入力 $a=(a_1,a_0)$，$b=(b_1,b_0)$，$c=(c_1,c_0)$ に対して，多数決の結果を $y=(y_1,y_0)$ として2ビット出力する「多数決結果出力回路」を考える．この多数決結果出力回路の各2ビット入力 a，b，c において，$(0,0)$ は入力組み合わせ禁止（ドントケア）となっており，各入力は $(0,1)$，$(1,0)$，あるいは $(1,1)$ の内，いずれかの値を取る．一方，この多数決結果出力回路の2ビット出力 y は2ビット入力 a，b，c の内，2つ以上の入力が同じ値を取るときにはその値が出力され，それ以外の場合は $y=(0,0)$ が出力される．例えば $a=(1,1)$，$b=(1,0)$，$c=(1,0)$ のとき $y=(1,0)$ が出力され，$a=(1,1)$，$b=(1,0)$，$c=(0,1)$ のときは $y=(0,0)$ が出力される．この多数決結果出力回路を2ビットデコーダ D，多数決回路 M，2ビットエンコーダ E を接続することにより実現する．以下の問いに答えよ．

(i) 2ビットデコーダ D は2ビット入力 $p=(p_1,p_0)$ と3ビット出力 $q=(q_3,q_2,q_1)$ をもつ．ただし $p=(0,0)$ は入力組み合わせ禁止である．p の10進数表現が i $(i=1,2,3)$ のとき $q_i=1$ となり，それ以外の場合は0となる．すなわち D の入出力関係は次の表で与えられる．

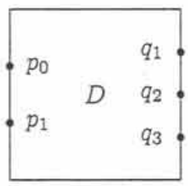
p_1	p_0	q_3	q_2	q_1
0	0	—	—	—
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

入力組み合わせ禁止に注意して，q_3，q_2，q_1 をなるべく簡単化された p_1，p_0 の論理式で表せ．

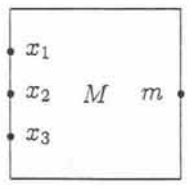
(ii) 多数決回路 M は3つの1ビット入力 x_1，x_2，x_3 と1ビット出力 m をもつ．m の値は x_1，x_2，x_3 の内，2つ以上の入力が1である場合は1，それ以外は0である．m を x_1，x_2，x_3 の最も簡単化された積和形で表せ．また，この多数決回路 M をNANDゲートのみを用いて実現し，図示せよ．ただし，NANDゲートの入力数に制限はないものとする．

(iii) 2ビットエンコーダ E は3ビット入力 $q=(q_3,q_2,q_1)$ と2ビット出力 $p=(p_1,p_0)$ をもつ．ただし，3ビット入力 q において複数の q_i が1となることはなく，これらは全て入力組み合わせ禁止とする．2ビット出力 p は $q_i=1$ のとき i の2進数表現となる．すなわち，$q_3=1$ ならば $p=(1,1)$，$q_2=1$ ならば $p=(1,0)$，$q_1=1$ ならば $p=(0,1)$ である．また，全ての $i=1,2,3$ に対して $q_i=0$ ならば $p=(0,0)$ となる．入力組み合わせ禁止に注意して，p_0，p_1 をなるべく簡単化された q_1，q_2，q_3 の論理式で表せ．

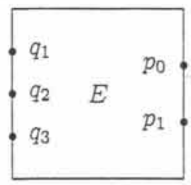
(iv) 3つの2ビットデコーダ D，3つの多数決回路 M，ならびに1つの2ビットエンコーダ E を接続して，この問題の最初で述べた，3つの2ビット入力 $a=(a_1,a_0)$，$b=(b_1,b_0)$，$c=(c_1,c_0)$ と2ビット出力 $y=(y_1,y_0)$ をもつ多数決結果出力回路を構成し，図示せよ．ただし，問い(i)，(ii)，(iii)で設計した3種の回路 D，M，E に対して，それぞれ下記の記号を用いること．



2ビットデコーダ D



多数決回路 M



2ビットエンコーダ E

基本アルゴリズム・プログラミング

7－1　2 分探索木とは，2 分木の各節点にデータを対応づけた上で，「ある節点のデータを x とするとき，その左部分木（左の子を根とする部分木）内のデータはすべて x より小さく，右部分木内のデータはすべて x より大きい」という条件を満たす 2 分木を指すものとする．この 2 分探索木について，以下の問いに答えよ．

(i)　高さ h の 2 分探索木において，節点数（ノード数）の最小値および最大値を答えよ．

(ii)　次の昇順に並んだ 9 個のデータについて，データ探索の計算量が最も少なくなる 2 分探索木（最適 2 分探索木）を一つ図示せよ．また，逆に計算量が最も多くなる 2 分探索木（最悪 2 分探索木）を一つ図示せよ．

(1,3,5,9,12,15,17,21,24)

(iii)　プログラム A は，データ集合を順に 2 分探索木に収納する関数 insert と，あるデータ（ここでは「9」を設定）が 2 分探索木に含まれているかどうかを探索する関数 member を含む C 言語のプログラムである．プログラム A にある空欄(1)～(3)を埋めよ．

(iv) プログラム A において，問い(ii)で答えた最適 2 分探索木を構築するためには，プログラム A の＜α＞部分において配列 x[]をどのように初期化すべきか．問い(ii)で挙げた 9 個のデータの並び順を答えよ．

(v) 問い(iv)で初期化された配列 x[ ]に対してプログラム A を実行したとき，member 関数における printf 文の出力結果を全て示せ．ただし，プログラム A の実行が完了するまでの出力を順に示すこと．

＜プログラム A＞

```
#include <stdio.h>
#include <stdlib.h>
struct node{
  int element;
  struct node *left;
  struct node *right;
};

char member(int x, struct node *p)
{
   struct node *q;
   q=p;
   while(q != NULL)
   {
       if(q->element == x) return('Y'); /** Yes **/
       if(q->element < x)
            {printf("%2d : right¥n",q->element); q = q->right;}
       else
            {printf("%2d : left¥n",q->element); q = q->left;}
    }
   return('N'); /** No **/
}

void insert(int x, struct node **p)
{
   struct node *q, **r;
   r=p; q=*r;
   while(q != NULL)
   {
     if(q->element == x) return;
     if(q->element < x) { r=&(    (1)    ); q=q->right;}
     else { r=&(    (2)    ); q=q->left;}
   }
   *r = (struct node *)malloc(sizeof(struct node));
   (*r)->element =      (3)      ;
   (*r)->left = NULL;
   (*r)->right = NULL;
   if(*p == NULL) *p = *r;
   return;
 }

int main()
{
  struct node *init;
  int x[]={          <α>          };
  int i;   char a;
  init = NULL;
  for(i=0;i<9;i++) insert(x[i], &init);
  a=member(9, init);
  printf("(Y/N) = %c¥n",a);
  return(0);
}
```

7－2　プログラム B は，ある与えられた配列を前半部と後半部に分け，それぞれをソートした後，マージ（併合）することにより元の配列のソートを行うマージソートの C 言語プログラムである．以下の問いに答えよ．

(i) プログラム B にある空欄(1)～(3)を埋めよ．

(ii) プログラム B はデータを昇順にソートするプログラムであるが，一行を書き換えて降順にソートするプログラムに変更せよ．ただし，該当する一行の変更前と変更後を記せ．

(iii) プログラム B（昇順ソート）において，merge_main 関数における printf 文の出力結果をプログラム B の実行が完了するまで順に示せ．

<プログラム B>

```
#include  <stdio.h>
#include  <stdlib.h>
int    *buff;
void merge_main(int a[], int left, int right)
{
        int     i,j=0,k=left,p=0,center = (left + right) / 2;
        if (left < right) {
                merge_main(a,[   (1)   ], center);
                merge_main(a,[   (2)   ], right);
                for (i = left; i <= center; i++)
                      buff[p++] = a[i];/* a[]の前半部を buff[]へ代入 */

                while (i <= [   (3)   ] && j < p )
                      a[k++] = (buff[j] <= a[i]) ? buff[j++] : a[i++];
                                          /* a[]の後半部と buff[]を比較 */

                while ( j < p )
                      a[k++] = buff[j++]; /* buff[]の残りを a[]へ代入 */

                printf("L = %d  R = %d¥n",left,right);
        }
        return;
}
int mergesort(int a[], int n)
{
        if((buff = (int *)calloc(n,sizeof(int)))==NULL) return(-1);
        merge_main(a, 0, n - 1);
        free(buff);
        return(0);
}
int main()
{
        int     i,x[7]={34,23,54,12,3,56,78};
        int     nx = sizeof(x) / sizeof(x[0]);
        if(mergesort(x, nx) == -1){printf("Error"); return(1);}
        for (i = 0; i < nx; i++) printf("x[%d] = %d¥n", i, x[i]);
        return(0);
}
```