

問1 次の(1)~(2)の問いに答えなさい。

(1) 次の①~⑤の式を計算する仮想計算機コードを示しなさい (結果を表示する命令コードは不要)。

① $1 - 2$ ② $1 * 2$ ③ $1 + 2 * 3$ ④ $1 * 2 + 3$ ⑤ $1 * (2 - 3)$

(2) 次の①~⑤の仮想計算機コードの実行結果 (スタックトップに残る値) を示しなさい。

①	②	③	④	⑤
pushi 3	pushi 3	pushi 1	pushi 1	pushi 3
pushi 4	pushi 4	pushi 3	pushi 2	pushi 1
add	sub	pushi 4	add	ne
		add	pushi 3	
		mul	mul	

問2 次のC言語のプログラムについて下の問いに答えなさい。

```
int a;
f(){
    int b;
    static int c; ...
}
```

(1) 伝統的な UNIX システムにおいて変数 a, b, c が配置される領域を①~④から選択しなさい。

①テキストセグメント ②データセグメント ③BSS ④スタックセグメント

(2) 関数呼び出し時の戻り番地が置かれる領域を上の①~④から選択しなさい。

(3) 機械語命令が置かれる領域を上の①~④から選択しなさい。

(4) malloc 等で確保される動的データが置かれる領域を上の①~④から選択しなさい。

問3 次の(1)~(2)の問いに答えなさい。

(1) 識別子の登録数を n とするとき検索の手間が次の(a)~(c)になる 識別子表の構成法 (データ構造) を答えなさい

(a) 検索の手間は識別子数に比例する

(b) 検索の手間は識別子の出現順序がランダムであれば $\log_2 n$ に比例する

(c) 表の混雑度 (占有率) が一定以下の場合, n によらず検索の手間は定数になる

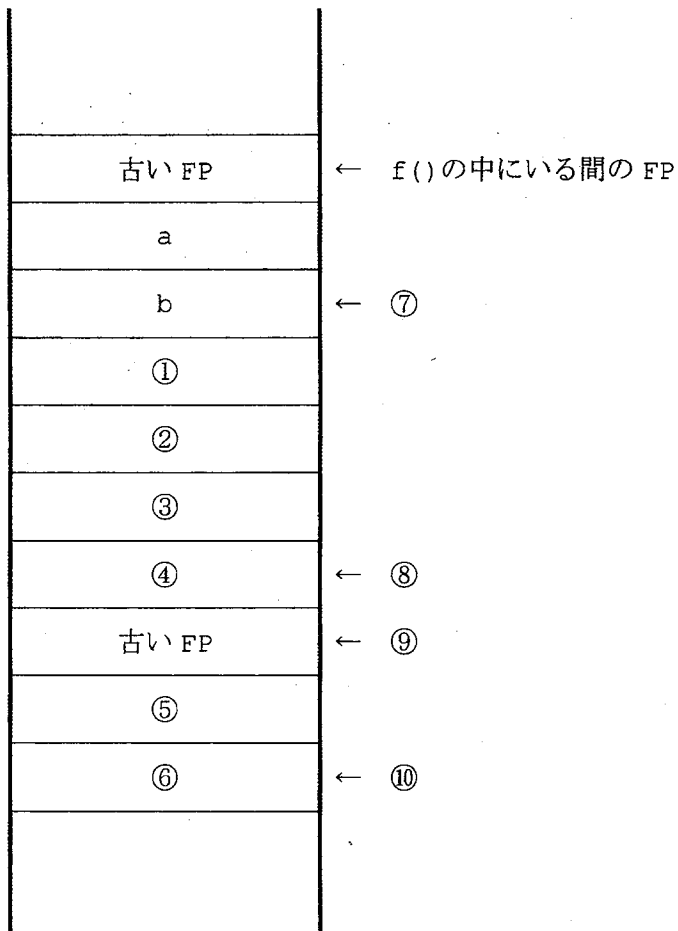
(2) (c)のうち, 識別子の衝突が起こった場合に別の計算式によって格納場所を再計算する方式の名前を答えなさい。

問4 picoc09 において下のプログラムを実行し、関数 f()から呼ばれた関数 g()の最初の文（下線で示した行）を実行する直前であるとする。①～⑩の空欄を(a)～(j)の語句で補い、スタックの状態を完成させなさい。

```
int f(){
    int a,b;
    a = 1;
    b = 2;
    a = g(a,b);
}

int g(int x,int y){
    int i,j;
    i = x + y;
    return (i);
}
```

- (a) f()の最初の文実行前の SP (b) f()への戻り番地 (c) i (d) j
 (e) g()が呼び出されたときの SP (f) g()の中にいる間の FP (g) g()からの戻り値
 (h) g()の最初の文実行前の SP (i) 1 (j) 2



picoc09 仮想マシンの命令一覧

アセンブラ表記	意味	動作
pushl <i>n</i>	ローカル変数をプッシュ	push(*(fp + n));
storel <i>n</i>	ローカル変数への格納	*(fp + n) = *sp;
storet <i>n</i>	一時領域への格納	*(sp + n) = *sp;
pushi <i>d</i>	定数をプッシュする	push(d);
call <i>label</i>	関数を呼び出す	push(pc); pc = label;
ret	関数から復帰する	pc = pop();
enter	スタックフレームを生成	push(fp); fp = sp;
leave	スタックフレームを解放	sp = fp; fp = pop();
mvsp <i>n</i>	SP を移動する	sp = sp + n;
jp <i>label</i>	無条件に分岐する	pc = label;
jt <i>label</i>	真なら分岐する	if (pop() != 0) pc = label;
jf <i>label</i>	偽なら分岐する	if (pop() == 0) pc = label;
add (sub, mul, div, mod)	加算を行う (減 / 乗 / 除 / 剰余)	t2 = pop(); t1 = pop(); push(t1 + t2);
le (eq, ne, gt, ge, lt)	「<=」 のとき真 (==, !=, >, >=, <)	t2 = pop(); t1 = pop(); if (t1 <= t2) push(1); else push(0);
rd	値の読み込み	scanf("%d",&t); push(t);
wr	値の出力	t = pop(); printf("%d ", t);
wrln	改行	printf("\n");
halt	プログラムの停止	

※*n*, *d*, *label* はオペランド

※pc, sp, fp はそれぞれプログラムカウンタ, スタックポインタとフレームポインタ
(仮想 CPU のレジスタ)

※t, t1, t2 などは説明のための便宜的な変数

※push, pop の動作は次のとおりである。

```

push(x) {
    sp--;
    *sp = x;
}

pop() {
    t = *sp;
    sp++;
    return t;
}

```

学籍番号 J

問1

(1)

① pushi 1
pushi 2
sub

② pushi 1
pushi 2
mul

③ pushi 1
pushi 2
pushi 3
mul
add

④ pushi 1
pushi 2
mul
pushi 3
add

⑤ pushi 1
pushi 2
pushi 3
sub
mul

(2) ① 7 ② -1 ③ 7 ④ 9 ⑤ 0

問2

(1) ② ④ ① ③

問3

a: ④ b: ② c: ④

(1) (a) リニアリスト (b) 順序木 (c) 木構造ハッシュ表

(2) オープンアドレスハッシュ

問4

① (1) ② (1) ③ (9) ④ (b) ⑤ (c)
⑥ (d) ⑦ (a) ⑧ (e) ⑨ (f) ⑩ (h)