問題　次のプログラムの説明を読み，プログラム中の［　　］に入れるべき適切な字句を解答群から選べ。ここで，配列の要素番号は0から始まる。

［プログラムの説明］

異なる数値が昇順に格納されている配列dataの中から，変数Xと同じ数値が格納されている要素を２分探索法を用いて探し，その要素を配列dataから削除するプログラムBinary_sである。なお，変数d_lenには配列dataの要素数が格納されている。

［手順］

① 探索範囲の先頭要素の添字をL，末尾要素の添字をHとする。なお，初期値は，Lは0，Hはd_len－1である。

② 探索範囲の中央要素となるdata[M]と比較する。ただし，M ← (L＋H)÷2とし，小数点以下は切り捨てる。

data[M]＜Xなら，L ← M＋1とし，次の探索範囲を，配列の要素位置がMより大きい方とする。

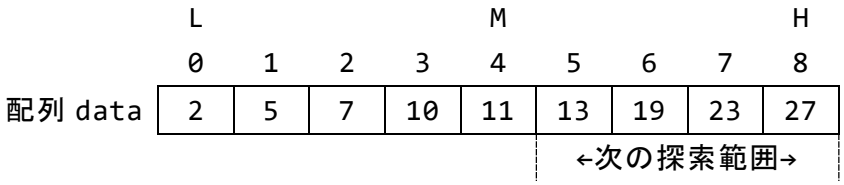


図１　比較例１

data[M]＞Xなら，H ← M－1とし，次の探索範囲を，配列の要素位置がMより小さい方とする。

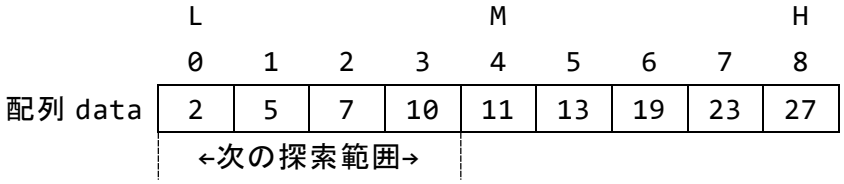
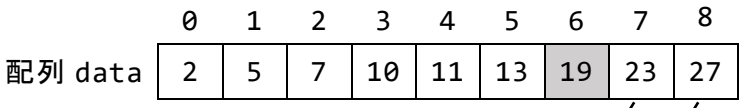


図２　比較例２

③ 変数Xと同じ数値が見つかった場合，その要素を配列dataから削除し，当該要素以降の要素を順に１つずつ前に移動する。また，変数d_lenの値を１減らす。例えば，配列dataの内容が図１と同じ状態で，変数d_len＝9，変数X＝19の場合，変数X＝19と同じ数値が配列data[6]に存在したため，配列data[7]以降の要素を順に１つずつ前に移動し，変数d_lenを８とする。

実行前



実行後

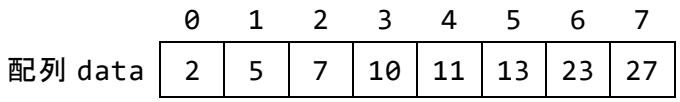


図３　要素の削除例

④ 変数 X と同じ数値がなかった場合，エラーメッセージを表示する。

［擬似言語の記述形式の説明］

記述形式	説明
○*手続名又は関数名*	手続又は関数を宣言する。
型名: *変数名*	変数を宣言する。
/* *注釈* */	注釈を記述する。
// *注釈*	
変数名 ← *式*	変数に*式*の値を代入する。
手続名又は関数名(*引数*, …)	手続又は関数を呼び出し，*引数*を受け渡す。

if (*条件式1*)
　処理1
elseif (*条件式2*)
　処理2
elseif (*条件式n*)
　処理n
else
　処理n＋1
endif
	選択処理を示す。
	*条件式*を上から評価し，最初に真になった*条件式*に対応する*処理*を実行する。以降の*条件式*は評価せず，対応する*処理*も実行しない。どの*条件式*も真にならないときは，*処理n＋1*を実行する。
	各*処理*は，0 以上の文の集まりである。
	elseif と*処理*の組みは，複数記述することがあり，省略することもある。
	else と*処理n＋1* の組みは一つだけ記述し，省略することもある。

while (*条件式*)
　処理
endwhile
	前判定繰返し処理を示す。
	*条件式*が真の間，*処理*を繰返し実行する。
	*処理*は，0 以上の文の集まりである。

do 　*処理* while (*条件式*)	後判定繰返し処理を示す。 *処*理を実行し，*条件式*が真の間，*処理*を繰返し実行する。 *処理*は，0 以上の文の集まりである。

for (*制御記述*)
　処理
endfor
	繰返し処理を示す。
	*制御記述*の内容に基づいて，*処理*を繰返し実行する。
	*処理*は，0 以上の文の集まりである。

〔演算子と優先順位〕

演算子の種類		演算子	優先度
式		( ) .	高
単項演算子		not ＋ －	↑
二項演算子	乗除	mod × ÷	↑
	加減	＋ －	↑
	関係	≠ ≦ ≧ ＜ ＝ ＞	↑
	論理積	and	↑
	論理和	or	低

注記 演算子 . は，メンバ変数又はメソッドのアクセスを表す。
演算子 mod は，剰余算を表す。

〔論理型の定数〕
true, false
〔配列〕
配列の要素は，″[″ と ″]″ の間にアクセス対象要素の要素番号を指定することでアクセスする。なお，二次元配列の要素番号は，行番号，列番号の順に ″, ″ で区切って指定する。
″{″ は配列の内容の始まりを，″}″ は配列の内容の終わりを表す。ただし，二次元配列において，内側の ″{″ と ″}″ に囲まれた部分は，1 行分の内容を表す。
〔未定義，未定義の値〕
変数に値が格納されていない状態を，″未定義″ という。変数に ″未定義の値″ を代入すると，その変数は未定義になる。

[プログラム]

```
○Binary_s (整数型：data[], 整数型：d_len, 整数型：X)
  整数型：L, H, M, p
  L ← 0
  H ← d_len − 1
  M ← (L + H) ÷ 2          /* 小数点以下は切り捨てる */
  /* 配列の中から X を探索する */
  while( L ≦ H and data[M] ≠ X )    ←――――   α
      if( data[M] > X )
          (1)
      else
          (2)
      endif
      M ← (L + H) ÷ 2      /* 小数点以下は切り捨てる */
  endwhile
  if( L ≦ H )
      (3)
      while( p < d_len )
      data[p−1] ← data[p]  /* データの削除処理 */
          (4)
      endwhile
      d_len ← d_len − 1
  else
      エラーメッセージを表示する
  endif
```

＜設問１＞　プログラム中の［　　］に入れるべき適切な字句を解答群から選べ。

(1)，(2) の解答群

ア．H ← M － 1　　イ．H ← M ＋ 1

ウ．L ← M － 1　　エ．L ← M ＋ 1

(3)，(4) の解答群

ア．p ← M － 1　　イ．p ← M ＋ 1

ウ．p ← p － 1　　エ．p ← p ＋ 1

＜設問２＞　配列dataの内容が次のような場合，プログラム中のαを実行するときの変数L，H，Mをトレースした表の［　　］に入れるべき適切な字句を解答群から選べ。

X | 42

	0	1	2	3	4	5	6	7	8	9
data	2	5	9	10	15	23	29	33	37	42

表　トレースの内容

順番	L	H	M
1	0	9	4
2	5	9	7
3	(5)		
4	9	9	9

(5) の解答群

	L	H	M
ア.	5	7	6
イ.	6	7	6
ウ.	7	9	8
エ.	8	9	8

	(1)	(2)	(3)	(4)	(5)
解答	ア	エ	イ	エ	エ