

次の問 8 は必須問題です。必ず解答してください。

問 8 次のプログラムの説明及びプログラムを読んで、設問に答えよ。

N 個の要素中から K 個の要素を選ぶ組合せをすべて求める。例えば、5 個の要素中から 3 個の要素を選ぶ組合せの場合、計 10 通りある組合せをすべて求める。

プログラムでは、N 個の要素（要素番号 1～N）からなる配列 S を用意し、このうち K 個の要素には 1 を、残りの要素には 0 を設定することによって、組合せの一つを表現する。例えば、図 1(1) のように 5 個の要素 1～5 中から 3 個の要素 2, 4, 5 を選んだ状態は、プログラム中では図 1(2) のとおりに表現する。



図 1 5 個の要素中から 3 個の要素を選ぶ例とそのプログラム中での表現

〔プログラムの説明〕

プログラムは、主プログラム Main 並びに組合せを求めるための関数 Init 及び Next からなる。

主プログラム Main

機能：N=5, K=3 として、5 個の要素中から 3 個の要素を選ぶ組合せ計 10 通りを順次求めて、配列 S に設定する。

整数型関数：Init(整数型：S[], 整数型：N, 整数型：K)

引数：S[] は出力用、N 及び K は入力用の引数である。

機能：1 ≤ K ≤ N の場合、配列 S の先頭から K 個の要素に 1 を、続く N - K 個の要素に 0 をそれぞれ設定し、返却値として 0 を返す。それ以外の場合、配列 S には値を設定せずに、返却値として -1 を返す。

整数型関数：Next(整数型：S[], 整数型：N)

引数：S[] は入出力用、N は入力用の引数である。

機能：渡された配列 S の先頭から N 個の要素には、直前に求めた組合せの状態が設定されている。この渡された組合せの状態に対して所定の操作を行い、次の組合せの状態を求めて配列 S に設定し、返却値として 0 を返す。ただし、渡された組合せの状態が、この関数のアルゴリズムで得られる最終形である場合、配列 S には値を設定せずに、返却値として -1 を返す。

[プログラム]

○主プログラム: Main

○整数型: $S[5]$, K , N , R

• $K \leftarrow 3$

• $N \leftarrow 5$

• $R \leftarrow \text{Init}(S, N, K)$

■ $R = 0$

• $R \leftarrow \text{Next}(S, N)$

■

/* $1 \leq K \leq N$ */

/* 選択する要素の個数 */

/* 要素の個数 */

← α

○整数型関数: $\text{Init}(\text{整数型: } S[], \text{整数型: } N, \text{整数型: } K)$

○整数型: L

▲ $1 \leq K \text{ and } K \leq N$

■ $L: 1, L \leq N, 1$

▲ $L \leq K$

• $S[L] \leftarrow 1$

• $S[L] \leftarrow 0$

■

• return 0

• return -1

○整数型関数: $\text{Next}(\text{整数型: } S[], \text{整数型: } N)$

○整数型: C, L, R

• $C \leftarrow 0$

• $L \leftarrow 1$

• $R \leftarrow -1$

■ $L < N \text{ and } R = -1$

▲ $S[L] = 1$

▲ $S[L+1] = 0$

• $S[L] \leftarrow 0$

• $S[L+1] \leftarrow 1$

• $\text{Init}(S, L-1, C)$

• $R \leftarrow 0$

• $C \leftarrow C + 1$

• $L \leftarrow L + 1$

■

• return R

設問 次の記述中の に入れる正しい答えを、解答群の中から選べ。

- (1) 主プログラム Main で、配列 S に組合せの一つの状態が得られるたびに、配列 S の内容を印字したい。印字には次の副プログラムを用いる。

副プログラム Dump(整数型: S[], 整数型: N)

引数: S[] 及び N は入力用の引数である。

機能: 配列 S の先頭から N 個の要素に格納されている値を、1 行に印字する。

そのためには、主プログラム Main の α の部分を a に示す部分と入れ替えればよい。

- (2) 関数 Next は、受け取った配列 S を要素番号の小さい方から検査し、連続する 2 要素の値が b に見つかったものについて、その内容を入れ替える。続いて、配列 S の一部でその 2 要素 c の部分について関数 Init を呼ぶ。例えば、関数 Next の実行開始時点で、配列 S の要素番号 1~5 の内容が 1, 0, 1, 0, 1 であったとき、実行終了時点での配列 S の要素番号 1~5 の内容は d となる。
- (3) このプログラムを実行して、関数 Init が関数 Next から呼ばれるとき、関数 Init が受け取る N の値の範囲は e , K の値の範囲は f である。したがって、関数 Init が受け取る N と K の値は、 $1 \leq K \leq N$ を満たさない場合がある。
- (4) 主プログラム Main の実行終了時点において、配列 S の要素番号 1~5 の内容は g となっている。

aに関する解答群

- | | |
|---|---|
| <p>ア</p> <ul style="list-style-type: none"> • $R \leftarrow \text{Init}(S, N, K)$ ■ $R = \emptyset$ • $\text{Dump}(S, N)$ • $R \leftarrow \text{Next}(S, N)$ ■ | <p>イ</p> <ul style="list-style-type: none"> • $R \leftarrow \text{Init}(S, N, K)$ ■ $R = \emptyset$ • $R \leftarrow \text{Next}(S, N)$ • $\text{Dump}(S, N)$ ■ |
| <p>ウ</p> <ul style="list-style-type: none"> • $R \leftarrow \text{Init}(S, N, K)$ • $\text{Dump}(S, N)$ ■ $R = \emptyset$ • $R \leftarrow \text{Next}(S, N)$ • $\text{Dump}(S, N)$ ■ | <p>エ</p> <ul style="list-style-type: none"> • $R \leftarrow \text{Init}(S, N, K)$ ■ $R = \emptyset$ • $\text{Dump}(S, N)$ • $R \leftarrow \text{Next}(S, N)$ ■ • $\text{Dump}(S, N)$ |

bに関する解答群

- ア 0, 1で最後 イ 0, 1で最初 ウ 1, 0で最後 エ 1, 0で最初

cに関する解答群

- ア 及びその後 イ 及びその前 ウ より後 エ より前

dに関する解答群

- ア 0, 1, 1, 0, 1 イ 1, 0, 0, 1, 1 ウ 1, 0, 1, 1, 0 エ 1, 1, 0, 0, 1

e, fに関する解答群

- | | | | |
|-------|-------|-------|-------|
| ア 0～2 | イ 0～3 | ウ 1～3 | エ 1～4 |
| オ 2～4 | カ 2～5 | | |

gに関する解答群

- ア 0, 0, 0, 0, 0 イ 0, 0, 1, 1, 1 ウ 1, 1, 1, 0, 0 エ 1, 1, 1, 1, 1