

受 験 番 号	氏 名

○

○

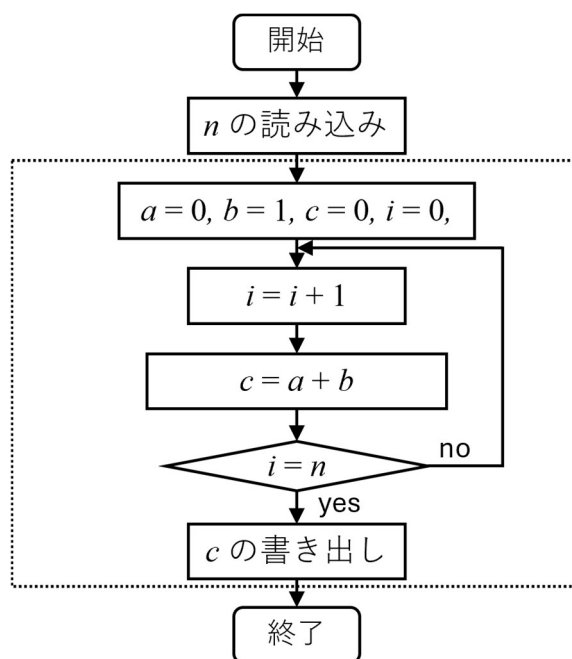
○

○

10. プログラミング	1/4
-------------	-----

(科目合計点)

1 (1)



(2)

```

#include<stdio.h>
int main(){
    int    i, n, a, b, c;
    scanf("%d",&n);
    a = 0; b = 1;
    for ( i=0; i<n; i++ )
        c = a + b; a=b; b=c;
    printf("%d¥n",c);
    return();
}

```

得点

受 験 番 号	氏 名

○

○

○

○

10. プログラミング	2/4
-------------	-----

2 (1)

```

int    weight(A[], WEIGHT[], int N){
    int    sum_weight=0;
    for ( int i=0; i<N; i++ )
        sum_weight = sum_weight + A[i]*WEIGHT[i];
    return(sum_weight);
}

int    value(A[], VALUE[], int N){
    int    sum_value=0;
    for ( int i=0; i<N; i++ )
        sum_value = sum_value + A[i]*VALUE[i];
    return(sum_value);
}

```

得点

受 験 番 号	氏 名

○

○

○

○

10. プログラミング	3/4
-------------	-----

2 (2)

```

#include <stdio.h>
#include <math.h>
#define MAX_N 100 // 品物の最大数
#define MAX_W 1000 // ナップサックの最大容量

int    weight(int A[], int WEIGHT[], int N){
    int    sum_weight=0;
    for ( int i=0; i<N; i++ )
        sum_weight = sum_weight + A[i]*WEIGHT[i];
    return(sum_weight);
}
int    value(int A[], int VALUE[], int N){
    int    sum_value=0;
    for ( int i=0; i<N; i++ )
        sum_value = sum_value + A[i]*VALUE[i];
    return(sum_value);
}
int    main(){
    int    N, W, max_value;
    int    A[N], max_A[N], WEIGHT[N], VALUE[N];
    printf("品物の数 N を入力: ");        scanf("%d", &N);
    printf("ナップサックの容量 W を入力: ");    scanf("%d", &W);
    for (int i=0; i < N; i++)
        max_A[i] = 0;

```

得点



(表面で答案が入りきらない場合は 以下に答案の続きを記入すること)

```

    for (int i = 0; i < N; i++) {
        printf("品物 %d の重さと価値を入力 (wi vi): ", i + 1);
        scanf("%d %d", &WEIGHT[i], &VALUE[i]);
    }
    for (int n=0; n<pow(2,N); n++){
        int    num = n;
        for (int m=N-1; m >= 0; m -- ){
            A[m] = num % 2;
            num /= 2;
        }
        int w = weight(A, WEIGHT, N);
        int v = value(A, VALUE, N);
        if ( w <= W && max_value < v ){
            max_value = v;
            for (int i=0; i<N; i++)
                max_A[i] = A[i];
        }
    }
    for (int i=0; i<N; i++)
        if (max_A[i] == 1) printf("%d ", i+1);
    return 0;
}

```

その他、動的計画法を用いるなども可

2 (3)

上記の全組合せを網羅する場合, $O(2^N)$

動的計画法を用いる場合, $O(N*W)$

受 験 番 号	氏 名

○

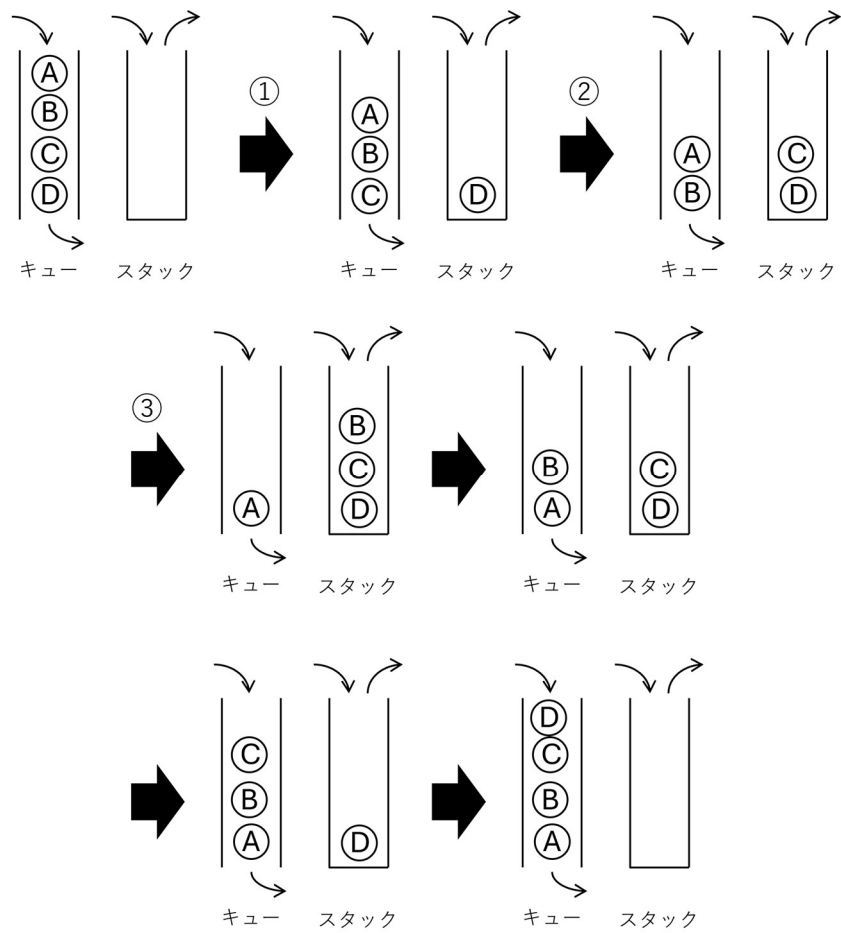
○

○

○

10. プログラミング	4/4
-------------	-----

3



3回

得点