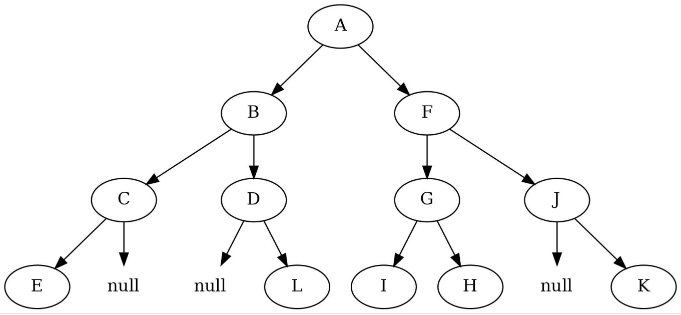
Algorithms and Data Structures

29.06.2023

Exercise 1 (1.5 points)

The following binary tree is given.



Visit it in post-order. Please, report the sequence of keys displayed by the visit on the same line and separate letters with a single space. No other symbols must be included in the response. This is an example of the response format: B D E A etc.

Exercise 2 (1.5 points)

A BST contains integer values included in the range 1-1000. Suppose we are looking for the value 531 is such a BST. Consider the following sequences of values generated during a search.

```
259 681 524 653 600 590 539 535 533 526 530 531
323 935 693 376 676 560 428 313 544 531
907 306 577 470 495 560 541 539 508 533 515 529 532 531
776 475 680 617 502 612 505 526 539 536 433 528 530 531
```

Indicate which ones of the following sequences (numbered 1, 2, 3, and 4) are correct. Notice that wrong responses imply a mark penalty.

Exercise 3 (1.5 points)

Suppose we have an initially empty priority queue implemented with a maximum heap. Consider the following sequence of integers and "*" characters, where each integer corresponds to one insertion into the priority queue and each character "*" corresponds to one extraction.

```
10  2  4 12  5  3 *  *
```

Report the sequence of values as they are stored in the array representing the priority queue at the end of the entire process. Please, show the entire content of the array as a sequence of integer values separated by a single space. No other symbols must be included in the response. This is an example of the response: 0 3 2 6 8 etc.

Exercise 4 (1.5 points)

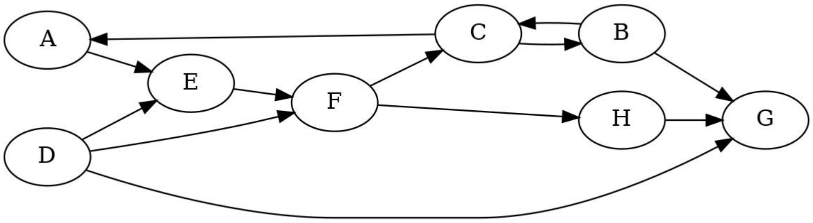
In an activity set, the i-th activity is identified by the pair $[s_i, f_i)$, where s_i is the starting time and f_i is the finishing time. The activities are numbered starting from 1. The following is a correct set of activities.

```
 P1  18  21
 P2   5   9
 P3  23  26
 P4   4   7
 P5   2   4
 P6  27  31
 P7  29  33
 P8  28  30
 P9  14  20
P10  30  34
```

Using a greedy algorithm, find the largest subset of mutually compatible activities. Please, report the set of compatible activities in the same order they have been selected, separated by a single space. No other symbols must be included in the response. This is an example of the response: 1 3 2 6 8

Exercise 5 (1.5 points)

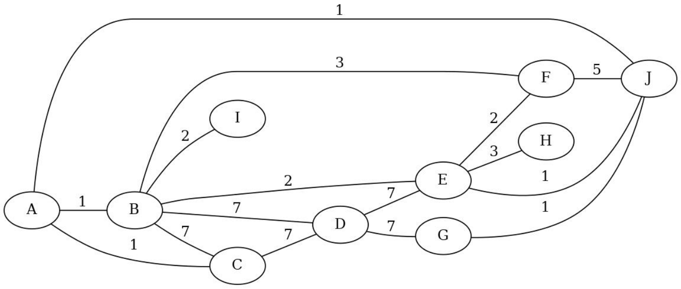
Given the following directed graph, represent the reverse graph and find all strongly connected components.



Compute all strongly connected components in the graph. Report them in alphabetic order and within each component indicates vertices in alphabetic order (i.e., ACZ and not CAZ or ZAC). Separate components with a single space. No other symbols must be included in the response. This is an example of the response format: ABC EF XY etc.

Exercise 6 (1.5 points)

Given the following undirected and weighted graph find a minimum spanning tree using Prim algorithm. Start from vertex A.



Indicate the total weight of the final minimum spanning tree. Report one single integer value. No other symbols must be included in the response. This is an example of the response format: 13

Exercise 7 (4.0 points)

Describe how to perform a Depth First Search on the graph G and report the pseudo-code of a standard DFS procedure. Moreover, define how to label edges (as Tree, Forward, Backward, and Cross) for directed.

Exercise 8 (2.0 points)

Analyze the following program and indicate the exact output it generates. Please, report the exact program output with no extra symbols.

```
int f (int [], int);
int main(void) {
  int i, l, v[32];
  l = f (v, 23);
  for (i=0; i<l; i++)
    printf ("%d", v[i]);
  return (1);
}
int f (int v[], int n) {
  int r, l;
  l = 0;
  while (n!=0) {
    r = n & 01;
    n = n >> 1;
    v[l++] = r;
  }
  return (l);
}
```

Exercise 9 (2.0 points)

The following function prints a powerset.

```
int powerset (int *val, int *sol, int n){
  int count, k;
  printf ("{ }\n");
  count = 0;
  for (k=1; k<=n; k++){ // LINE 1
    count += powerset_r (val,sol,n,k,0,0);
  }
  return count;
}
int powerset_r (int *val, int *sol, int n, int k, int start, int pos) {
  int count = 0, i;
  if (pos >= k){
    printf("{ ");
    for (i=0; i<k; i++)
      printf("%d ", sol[i]);
    printf(" }\n");
    return 1;
  }
  for (i=start; i<n; i++){
    sol[pos] = val[i]; // LINE 2
    count += powerset_2_r(val,sol,n,k,i,pos+1); // LINE 3
  }
  return count;
}
```

Indicates which ones of the following statements are correct. Note that incorrect answers imply a penalty in the final score.

1. Function powerset_r implements the simple combinations.
2. In LINE 3, the value i must be substituted by i+1.
3. Function powerset_r implements the arrangements with repetition principle.
4. Function powerset_r implements the combination with repetition principle.
5. In LINE 1, k must vary from 0 to n.
6. In LINE 3, the value pos+1 must be substituted by pos.
7. LINE 2 must be sol[i]=val[pos]

1.
2.
3.
4.
5.
6.
7.

Exercise 10 (2.0 points)

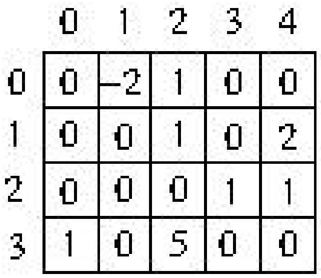
Analyze the following program and indicate the exact output it generates. Please, report the precise program output with no extra symbols.

```
int f (int, int);
int f (int n, int m) {
  int l1=0, l2=0;
  if (n<=0 && m<=0) {
    return 0;
  }
  if (n>0) {
    l1 = f (n-1, m);
  } else {
    if (m>0)
      l2 = f (n, m-1);
  }
  return (l1+l2+1);
}
int main () {
  int rv;
  rv = f(5,6);
  fprintf (stdout, "%d", rv);
  return 1;
}
```

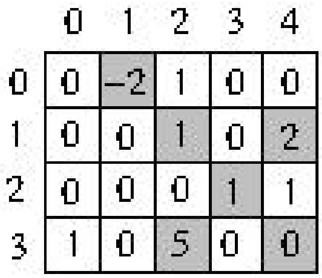
Exercise 11 (3.0 points)
A square matrix, with r rows and c columns, stores integer values.
Write the function

```
int max_diff (int **mat, int r, int c);
```

which receives the matrix mat and its size (r, c) and it displays the indices (row and column) of the element for which the difference between the maximum and minimum elements belonging to the diagonal and the inverse-diagonal is maximum.
For example, with $r = 4$, $c = 5$, and the matrix represented on the left-hand side of the following picture, we have to print the values $(2, 3)$ as represented on the right-hand side picture. For this value all elements of the diagonal and inverse diagonal are represented with a gray blackground. Amonhg all tese values the program must leect the maximum and the minum value, i.e., 5 and -2, and compèute the difference, i.e., $(5 - (-2)) = 7$. Write the entire program using standard C libraries but implement all required personal libraries. Modularize.tex the program adequately, and report a brief description of the data structure and the logic adopted in plain English. Unclear or awkward programs, complex or impossible to understand, will be penalized in terms of the final evaluation.



	0	1	2	3	4
0	0	-2	1	0	0
1	0	0	1	0	2
2	0	0	0	1	1
3	1	0	5	0	0

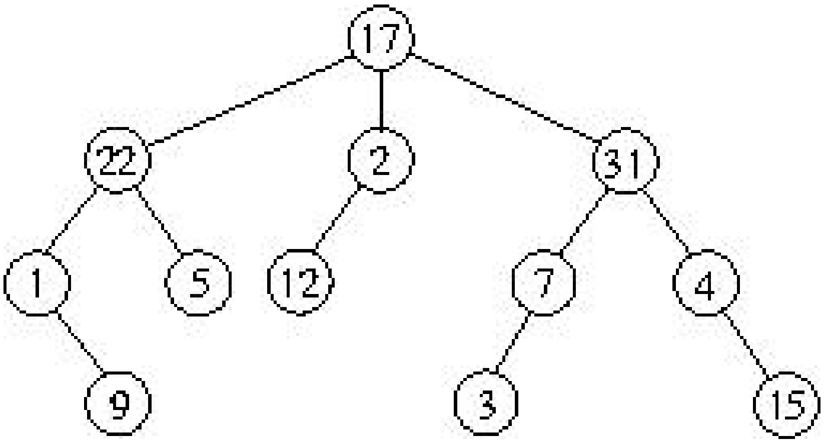


	0	1	2	3	4
0	0	-2	1	0	0
1	0	0	1	0	2
2	0	0	0	1	1
3	1	0	5	0	0

Exercise 12 (5.0 points)
An n-ary tree stores element with integer keys. We define as n the number of nodes at the same depth d in the tree (please, remind that in a tree the root has depth zero).
Write the function

```
int display_crowded_depth (node_t *root);
```

that receives the pointer to the root of such a tree and returns the depth value d for which n is maximum. The function must also display (in any order) all node keys at depth d. The candidate must define the node_t structure



For example, if the variable root points to the tree represented in the following picture the number of nodes n at depth 2 equals 5; it is the maximum number of nodes with the same depth. Thus, the function must display the following values (in any order) and returns 2. 1 5 12 7 4
Write the entire program using standard C libraries but implement all required personal libraries. Modularize the program adequately, and report a brief description of the data structure and the logic adopted in plain English. Unclear or awkward programs, complex or impossible to understand, will be penalized in terms of the final evaluation.

Exercise 13 (9.0 points)
Write a function to partition an array of integer values in a set of blocks satisfying some.texific properties.
The prototype of the function is the following one:

```
void partition (int *vet, int n, int k);
```

The function receives:

- An array on integer values vet.
- The size of the array n.
- An integer value k (smaller or equal to n).

and it partitions the n values stored in vet in k blocks (i.e., subsets) such that:

1. The number of values included in each block varies by at most one.
2. If Si is the sum of the values contained in each block, the variance of the Si values must be minimum (among all possible partitions of the values).

Please, remind that given k values Si their average and variance can be computed as:

$$A(S_i) = (S_1 + S_2 + ... + S_k)/k$$

$$V(S_i) = (\sum(S_i - mean(S_i))^2)/(k-1)$$

For example, let us suppose that vet = 1, 2, 3, 4, 5, n = 5, and k = 2. We have to generate all partitions of vet with k=2 blocks. A few of these partitions are the following: $\{(1),(2,3,4,5)\}$, $\{(1,2),(3,4,5)\}$, $\{(3,5),(1,2,4)\}$, etc. For the first one, i.e., $\{(1),(2,3,4,5)\}$, the first block has one element and the second block has four elements; thus, the partition does not satisfy constraint number 1 and must be discarded. For the last one, i.e., $\{(3,5),(1,2,4)\}$, $S_1 = (3+5) = 8$ and $S_2 = (1+2+4) = 7$, thus:

$$A(S_i) = (8+7)/2 = 7.5$$

$$V(S_i) = ((8-7.5)^2 + (7-7.5)^2)/(k-1) = (0.25+0.25)/1 = 0.5$$

The program must display the partition of the values in vet for which the value V(Si) is minimum.
Write the required function program using standard C libraries but implement all required personal libraries. Modularize the program adequately, and report a brief description of the data structure and the logic adopted in plain English. Unclear or awkward programs, complex or impossible to understand, will be penalized in terms of the final evaluation.