

Áttekintő:

- Verem implementálása (STACK) - dinamikusan
- Repository-k előkészítése
- Várakozási sor implementálása
- Egyszeresen láncolt lista implementálása

Kitűzött feladatok:

1. Feladat:

Implementáljuk a veremre (STACK) vonatkozó műveleteket, ezúttal dinamikus megvalósítást követve. Biztosítsd egy interaktív menü (switch case) használatát a Main-ben. Használd alapul az előadáson elhangzott információkat (19. dia ->):

https://moodle.ms.sapientia.ro/pluginfile.php/20160/mod_resource/content/2/6_Eloadas.pdf

Egy kis segítség a stack.h felépítéséhez:

```
typedef struct NodeType{
    int info;
    struct NodeType *next;
}NodeType;

bool isEmpty (NodeType* topPtr);

NodeType* create();

void push(NodeType** topPtr, int info);

void pop (NodeType** topPtr);

int top (NodeType* topPtr);

void printStack(NodeType* topPtr);
```

6. GYAKORLAT

2. Feladat:

Implementáljuk a várakozási sorra (QUEUE) vonatkozó műveleteket. Helyezd a várakozási sorra vonatkozó header és forrásfájlt ugyanabba a porjektbe. Használd alapul az előadáson elhangzott információkat. 34 dia ->

https://moodle.ms.sapientia.ro/pluginfile.php/20160/mod_resource/content/2/6_Eloadas.pdf

```
typedef struct ListItem
{
    int info;
    struct ListItem *next;
} ListItem;
```

Biztosíts egy interaktív menüt a Main-ben.

```
int main() {

    NodeType* myStack = NULL;
    int option;
    bool repeat = true;
    while(repeat) {
        printf("Lehetseges muveletek:\n"
            "0. Kilepes\n"
            "1. Elem hozzaadasa a veremhez\n"
            "2. Verem kiiratasa\n");

        printf("Valassz opciot: ");
        scanf("%i", &option);
        switch (option) {
            case 0:
                printf("Kilepes...");
                repeat = false;
                break;
            case 1: {
                ///push
                printf("Adj meg egy erteket: ");
                int item;
                scanf("%i", &item);
                push(&myStack,item);
                break;
            }
        }
    }
}
```

6. GYAKORLAT

```
    }
    case 2: {
        ///print stack
        printf("A verem tartalma: ");
        printStack(myStack);
        break;
    }
    default:
        printf("Ismeretlen opcio\n");
        break;
}
}

return 0;
}
```

3. Feladat:

Egészítsd ki úgy az előző feladatban megírt függvényeket, hogy a felhasználónak legyen lehetősége egyszeresen láncolt listát is létrehozni és azzal kapcsolatos műveleteket elvégezni. Pl: beszúr elejére

1. Extra:

Implementáld a duplán és körkörösén láncolt listára, valamint az elsőbbségi sorra vonatkozó műveleteket.

Hasznos linkek:

- <http://www.cprogrammingnotes.com/question/dynamic-stack.html>
- <http://www.cprogrammingnotes.com/question/dynamic-queue.html>
- <https://www.geeksforgeeks.org/queue-linked-list-implementation/>
- <https://www.geeksforgeeks.org/data-structures/linked-list/>