Töö nr.1 - mälu juhtimine
Erki Suurjaak
970772
LAP31

Programm simuleerib mälu töö juhtimist. Mäluna kasutatakse 64 märgi pikkust stringi, kus üks märk tähendab ühte mälublokki ja võimalikud väärtused on :
0 - Mälublokk on tühi ja kasutatav
* - Mälublokk sisaldab informatsiooni (* on suvaline sümbol (ASCII kood > 48)
# - Mälublokk on märgistatud prahina ja ei ole kasutatav

Mälupildi lõpus olev märk "-" tähendab, et mälu vabastati. "+" tähendab, et mälu reserveeriti.

Mälu reserveeritus kasvab umbes 80%-ini, kahaneb siis umbes 50%-ini ja kasvab uuesti umbes 75%-ini. Mälu reserveeritakse/vabastatakse iga variandi puhul umbes 72 korda.

Mälu täitmine toimub järgmiselt :
- mällu pandava elemendi maht on juhuslik suurus, võimalikud väärtused on 1..5 märki
- vastavalt algoritmile kas leitakse esimene sobiv blokk või valitakse kõigist võimalikest blokkidest sobivaim (s.o. kuhu element kõige täpsemalt ära mahub). Kui elementi ei ole võimalik ühe tükina mällu paigutada, siis ta fragmenteeritakse.

Mälu vabastamine toimub järgmiselt :
- valitakse suvaline mällu pandud element
- vastavalt algoritmile kas tehakse elemendi all olnud mälu kohe kasutuskõlblikuks või märgistatakse prahina; viimasel juhul koristatakse praht siis, kui mällu enam elemente ei mahu.

Variant #1 - mälu vabastamisel suunatakse vabastatud blokid vabade blokkide hulka ja mälu reserveerimisel leitakse esimene sobiv blokk

```
22220000000000000000000000000000000000000000000000000000000000000  +
22223330000000000000000000000000000000000000000000000000000000000  +
22223334444000000000000000000000000000000000000000000000000000000  +
22223330000000000000000000000000000000000000000000000000000000000  -
22223336600000000000000000000000000000000000000000000000000000000  +
22223336677000000000000000000000000000000000000000000000000000000  +
22223336677800000000000000000000000000000000000000000000000000000  +
22220006677800000000000000000000000000000000000000000000000000000  -
2222::0667780000000000000000000000000000000000000000000000000000  +
2222::;6677800000000000000000000000000000000000000000000000000000  +
2222::;66778<<0000000000000000000000000000000000000000000000000  +
2222::066778<<0000000000000000000000000000000000000000000000000  -
```

```
2222::066778<<>>0000000000000000000000000000000000000000000000000000  +
2222::066778<<>>??0000000000000000000000000000000000000000000000000  +
2222::066778<<>>??@@0000000000000000000000000000000000000000000000000  +
0000::066778<<>>??@@0000000000000000000000000000000000000000000000000  -
BB00::066778<<>>??@@0000000000000000000000000000000000000000000000000  +
BB00::066778<<>>??@@CCCC0000000000000000000000000000000000000000000000  +
BB00::066778<<>>??@@CCCCDDDD0000000000000000000000000000000000000000  +
BB00::066778<<>>??@@CCCC0000000000000000000000000000000000000000000  -
BB00::066778<<>>??@@CCCCFFFF0000000000000000000000000000000000000000  +
BBGG::066778<<>>??@@CCCCFFFF0000000000000000000000000000000000000000  +
BBGG::H66778<<>>??@@CCCCFFFF0000000000000000000000000000000000000000  +
BB00::H66778<<>>??@@CCCCFFFF0000000000000000000000000000000000000000  -
BB00::H66778<<>>??@@CCCCFFFFJJJ0000000000000000000000000000000000000  +
BB00::H66778<<>>??@@CCCCFFFFJJJKKKK0000000000000000000000000000000000  +
BBL0::H66778<<>>??@@CCCCFFFFJJJKKKK0000000000000000000000000000000000  +
00L0::H66778<<>>??@@CCCCFFFFJJJKKKK0000000000000000000000000000000000  -
00L0::H66778<<>>??@@CCCCFFFFJJJKKKKNNNN0000000000000000000000000000000  +
OOL0::H66778<<>>??@@CCCCFFFFJJJKKKKNNNN0000000000000000000000000000000  +
OOL0::H66778<<>>??@@CCCCFFFFJJJKKKKNNNNPPP0000000000000000000000000000  +
OOL0::H66778<<00??@@CCCCFFFFJJJKKKKNNNNPPP0000000000000000000000000000  -
OOL0::H66778<<RR??@@CCCCFFFFJJJKKKKNNNNPPP0000000000000000000000000000  +
OOL0::H66778<<RR??@@CCCCFFFFJJJKKKKNNNNPPPSS0000000000000000000000000  +
OOL0::H66778<<RR??@@CCCCFFFFJJJKKKKNNNNPPPSSTTT0000000000000000000000  +
OOL0::H6677800RR??@@CCCCFFFFJJJKKKKNNNNPPPSSTTT0000000000000000000000  -
OOL0::H6677800RR??@@CCCCFFFFJJJKKKKNNNNPPPSSTTTVVVV0000000000000000  +
OOL0::H66778WWRR??@@CCCCFFFFJJJKKKKNNNNPPPSSTTTVVVV0000000000000000  +
OOL0::H66778WWRR??@@CCCCFFFFJJJKKKKNNNNPPPSSTTTVVVVXXX0000000000000  +
OOL0::H66778WWRR??@@CCCCFFFFJJJKKKKNNNNPPPSSTTTVVVVXXXYY00000000000  +
OOL0::H66778WWRR??@@CCCC0000JJJKKKKNNNNPPPSSTTTVVVVXXXYY00000000000  -
OOL0::H66778WWRR??@@CCCC0000JJJKKKKNNNNPPPSSTTTVVVVXXX0000000000000  -
OOL0::H6677800RR??@@CCCC0000JJJKKKKNNNNPPPSSTTTVVVVXXX0000000000000  -
OOL]::H6677800RR??@@CCCC0000JJJKKKKNNNNPPPSSTTTVVVVXXX0000000000000  +
OOL]::H6677800RR??@@CCCC0000JJJKKKK0000PPPSSTTTVVVVXXX0000000000000  -
OOL]::H667780000??@@CCCC0000JJJKKKK0000PPPSSTTTVVVVXXX0000000000000  -
OOL]::H007780000??@@CCCC0000JJJKKKK0000PPPSSTTTVVVVXXX0000000000000  -
OOL]::Haa7780000??@@CCCC0000JJJKKKK0000PPPSSTTTVVVVXXX0000000000000  +
OOL]::Haa7780000??@@CCCC0000JJJKKKK0000PPP00TTTVVVVXXX0000000000000  -
OOL]::Haa7780000000@@CCCC0000JJJKKKK0000PPP00TTTVVVVXXX0000000000000  -
OO0]::Haa7780000000@@CCCC0000JJJKKKK0000PPP00TTTVVVVXXX0000000000000  -
OOe]::Haa7780000000@@CCCC0000JJJKKKK0000PPP00TTTVVVVXXX0000000000000  +
OOe]::0aa7780000000@@CCCC0000JJJKKKK0000PPP00TTTVVVVXXX0000000000000  -
OOe0::0aa7780000000@@CCCC0000JJJKKKK0000PPP00TTTVVVVXXX0000000000000  -
OOe0::0aa7780000000@@00000000JJJKKKK0000PPP00TTTVVVVXXX0000000000000  -
OOe0::0aa778ii0000@@00000000JJJKKKK0000PPP00TTTVVVVXXX0000000000000  +
OOe0::0aa778ii0000@@00000000JJJKKKK0000PPP00000VVVVXXX0000000000000  -
OOe0::0aa778iikkk0@@00000000JJJKKKK0000PPP00000VVVVXXX0000000000000  +
OOel::0aa778iikkk0@@00000000JJJKKKK0000PPP00000VVVVXXX0000000000000  +
OOel::0aa778iikkk0@@00000000JJJKKKK0000000000000VVVVXXX0000000000000  -
OOel::0aa778iikkk0@@nn000000JJJKKKK0000000000000VVVVXXX0000000000000  +
OOel::0aa778iikkk0@@nnoooo00JJJKKKK0000000000000VVVVXXX0000000000000  +
OOel::0aa778iikkk0@@nnoooo00JJJKKKKppp0000000000VVVVXXX0000000000000  +
OOel::0aa778iikkk0@@nnoooo00JJJKKKKppp0000000000000XXX0000000000000  -
OOel::0aa778iikkk0@@nnoooo00JJJKKKKpppprrrr000000000XXX0000000000000  +
OOel::0aa778iikkk0@@nnoooo00JJJKKKKpppprrrrssss00000XXX0000000000000  +
OOel::taa778iikkk0@@nnoooo00JJJKKKKpppprrrrssss00000XXX0000000000000  +
OOel::taa778iikkk0@@nnoooo00JJJKKKKppp0000ssss00000XXX0000000000000  -
OOel::taa778iikkkv@@nnoooo00JJJKKKKppp0000ssss00000XXX0000000000000  +
OOel::taa778iikkkv@@nnoooow0JJJKKKKppp0000ssss00000XXX0000000000000  +
OOel::taa778iikkkv@@nnoooow0JJJKKKKpppxxx0ssss00000XXX0000000000000  +
OOel::taa778iikkkv@@nnoooo00JJJKKKKpppxxx0ssss00000XXX0000000000000  -
OOel::taa778iikkkv@@nnooooz0JJJKKKKpppxxx0ssss00000XXX0000000000000  +
OOel::taa778iikkkv@@nnooooz{JJJKKKKpppxxx0ssss00000XXX0000000000000  +
```

Variant #1 - mälu vabastamisel suunatakse vabastatud blokid vabade blokkide hulka ja mälu reserveerimisel leitakse kõige sobivam blokk

```
2000000000000000000000000000000000000000000000000000000000000000000000  +
2300000000000000000000000000000000000000000000000000000000000000000000  +
2340000000000000000000000000000000000000000000000000000000000000000000  +
2300000000000000000000000000000000000000000000000000000000000000000000  -
2360000000000000000000000000000000000000000000000000000000000000000000  +
2367777000000000000000000000000000000000000000000000000000000000000000  +
2367777888000000000000000000000000000000000000000000000000000000000000  +
0367777888000000000000000000000000000000000000000000000000000000000000  -
0367777888::::0000000000000000000000000000000000000000000000000000000  +
0367777888::::;;;0000000000000000000000000000000000000000000000000000  +
0367777888::::;;;<<0000000000000000000000000000000000000000000000000  +
0367777000::::;;;<<0000000000000000000000000000000000000000000000000  -
0367777>>>::::;;;<<0000000000000000000000000000000000000000000000000  +
0367777>>>::::;;;<<????0000000000000000000000000000000000000000000000  +
0367777>>>::::;;;<<????@@@0000000000000000000000000000000000000000000  +
0367777>>>::::;;;00????@@@0000000000000000000000000000000000000000000  -
0367777>>>::::;;;00????@@@BBB0000000000000000000000000000000000000000  +
C367777>>>::::;;;00????@@@BBB0000000000000000000000000000000000000000  +
C367777>>>::::;;;DD????@@@BBB0000000000000000000000000000000000000000  +
C307777>>>::::;;;DD????@@@BBB0000000000000000000000000000000000000000  -
C307777>>>::::;;;DD????@@@BBBFF000000000000000000000000000000000000000  +
C307777>>>::::;;;DD????@@@BBBFFGGG000000000000000000000000000000000000  +
C307777>>>::::;;;DD????@@@BBBFFGGGHH000000000000000000000000000000000  +
0307777>>>::::;;;DD????@@@BBBFFGGGHH000000000000000000000000000000000  -
0307777>>>::::;;;DD????@@@BBBFFGGGHHJJ0000000000000000000000000000000  +
0307777>>>::::;;;DD????@@@BBBFFGGGHHJJKK0000000000000000000000000000  +
0307777>>>::::;;;DD????@@@BBBFFGGGHHJJKKLLL0000000000000000000000000  +
0307777000::::;;;DD????@@@BBBFFGGGHHJJKKLLL0000000000000000000000000  -
0307777000::::;;;DD????@@@BBBFFGGGHHJJKKLLLNNNN0000000000000000000000  +
0307777000::::;;;DD????@@@BBBFFGGGHHJJKKLLLNNNNOOOO0000000000000000000  +
P307777000::::;;;DD????@@@BBBFFGGGHHJJKKLLLNNNNOOOO0000000000000000000  +
P307777000::::;;;00????@@@BBBFFGGGHHJJKKLLLNNNNOOOO0000000000000000000  -
P3R7777000::::;;;00????@@@BBBFFGGGHHJJKKLLLNNNNOOOO0000000000000000000  +
P3R7777SSS::::;;;00????@@@BBBFFGGGHHJJKKLLLNNNNOOOO0000000000000000000  +
P3R7777SSS::::;;;T0????@@@BBBFFGGGHHJJKKLLLNNNNOOOO0000000000000000000  +
P3R7777SSS::::;;;T0????@@@000FFGGGHHJJKKLLLNNNNOOOO0000000000000000000  -
P3R7777SSS::::;;;T0????@@@VVVFFGGGHHJJKKLLLNNNNOOOO0000000000000000000  +
P3R7777SSS::::;;;T0????@@@VVVFFGGGHHJJKKLLLNNNNOOOOWWWW000000000000000  +
P3R7777SSS::::;;;T0????@@@VVV00GGGHHJJKKLLLNNNNOOOOWWWW000000000000000  -
P3R7777SSS::::;;;TY????@@@VVV00GGGHHJJKKLLLNNNNOOOOWWWW000000000000000  +
P3R7777SSS::::;;;TY????@@@VVV00GGGHHJJKKLLLNNNNOOOO0000000000000000000  -
P3R7777SSS::::;;;TY????@@@VVV00GGGHHJJKK000NNNNOOOO0000000000000000000  -
P307777SSS::::;;;TY????@@@VVV00GGGHHJJKK000NNNNOOOO0000000000000000000  -
P307777SSS::::;;;TY????@@@VVV]]GGGHHJJKK000NNNNOOOO0000000000000000000  +
P307777SSS0000;;;TY????@@@VVV]]GGGHHJJKK000NNNNOOOO0000000000000000000  -
P307777SSS0000;;;TY????@@@VVV]]GGGHHJJ00000NNNNOOOO0000000000000000000  -
P007777SSS0000;;;TY????@@@VVV]]GGGHHJJ00000NNNNOOOO0000000000000000000  -
P007777SSSaaaa;;;TY????@@@VVV]]GGGHHJJ00000NNNNOOOO0000000000000000000  +
P007777SSSaaaa;;;TY????@@@VVV]]GGG00JJ00000NNNNOOOO0000000000000000000  -
P007777SSSaaaa;;;TY????@@@VVV]]GGG00JJ00000NNNN000000000000000000000  -
P007777SSSaaaa;;;TY0000@@@VVV]]GGG00JJ00000NNNN000000000000000000000  -
P007777SSSaaaa;;;TYeee0@@@VVV]]GGG00JJ00000NNNN000000000000000000000  +
P007777SSSaaaa000TYeee0@@@VVV]]GGG00JJ00000NNNN000000000000000000000  -
P007777SSSaaaa000TYeee0@@@VVV]]00000JJ00000NNNN000000000000000000000  -
Phh7777SSSaaaa000TYeee0@@@VVV]]00000JJ00000NNNN000000000000000000000  +
Phh7777SSS0000000TYeee0@@@VVV]]00000JJ00000NNNN000000000000000000000  -
Phh7777SSS0000000TYeeej@@@VVV]]00000JJ00000NNNN000000000000000000000  +
```

```
Phh7777SSS0000000TYeeej@@@VVV]]kkkk0JJ00000NNNN00000000000000000  +
Phh7777SSS0000000TYeeej@@@VVV]]kkkk0JJllll0NNNN00000000000000000  +
Phh7777SSS0000000TYeeej000VVV]]kkkk0JJllll0NNNN00000000000000000  -
Phh7777SSS0000000TYeeejnnnVVV]]kkkk0JJllll0NNNN00000000000000000  +
Phh7777SSSoo00000TYeeejnnnVVV]]kkkk0JJllll0NNNN00000000000000000  +
Phh7777SSSooppp00TYeeejnnnVVV]]kkkk0JJllll0NNNN00000000000000000  +
Phh7777SSSooppp00TYeeejnnnVVV]]kkkk0JJllll000000000000000000000  -
Phh7777SSSooppp00TYeeejnnnVVV]]kkkk0JJllllrrrr000000000000000000  +
Phh7777SSSooppppssTYeeejnnnVVV]]kkkk0JJllllrrrr000000000000000000  +
Phh7777SSSooppppssTYeeejnnnVVV]]kkkk0JJllllrrrrtttt00000000000000  +
```

Variant #3 - mälu vabastamisel märgistatakse vabastavad blokid prahina ja mälu reserveerimisel leitakse esimene sobiv blokk

```
22200000000000000000000000000000000000000000000000000000000000000000  +
22233000000000000000000000000000000000000000000000000000000000000000  +
22233400000000000000000000000000000000000000000000000000000000000000  +
222##40000000000000000000000000000000000000000000000000000000000000  -
222##46600000000000000000000000000000000000000000000000000000000000  +
222##46677000000000000000000000000000000000000000000000000000000000  +
222##46677888800000000000000000000000000000000000000000000000000000  +
222##46677####000000000000000000000000000000000000000000000000000000  -
222##46677####::::000000000000000000000000000000000000000000000000  +
222##46677####::::;000000000000000000000000000000000000000000000000  +
222##46677####::::;<000000000000000000000000000000000000000000000000  +
222##4##77####::::;<000000000000000000000000000000000000000000000000  -
222##4##77####::::;<>>000000000000000000000000000000000000000000000  +
222##4##77####::::;<>>????000000000000000000000000000000000000000000  +
222##4##77####::::;<>>????@@000000000000000000000000000000000000000  +
222##4##77####::::;<##????@@000000000000000000000000000000000000000  -
222##4##77####::::;<##????@@B00000000000000000000000000000000000000  +
222##4##77####::::;<##????@@BCCC0000000000000000000000000000000000  +
222##4##77####::::;<##????@@BCCCDDD000000000000000000000000000000  +
222##4##77####::::;<##????@@#CCCDDD000000000000000000000000000000  -
222##4##77####::::;<##????@@#CCCDDDFFF0000000000000000000000000000  +
222##4##77####::::;<##????@@#CCCDDDFFFGGGG0000000000000000000000  +
222##4##77####::::;<##????@@#CCCDDDFFFGGGGHHHH0000000000000000000  +
#####4##77####::::;<##????@@#CCCDDDFFFGGGGHHHH0000000000000000000  -
#####4##77####::::;<##????@@#CCCDDDFFFGGGGHHHHJ0000000000000000000  +
#####4##77####::::;<##????@@#CCCDDDFFFGGGGHHHHJKK000000000000000000  +
#####4##77####::::;<##????@@#CCCDDDFFFGGGGHHHHJKKLL00000000000000000  +
#####4##77####::::;<##????@@#CCC###FFFGGGGHHHHJKKLL00000000000000000  -
#####4##77####::::;<##????@@#CCC###FFFGGGGHHHHJKKLLNNNN000000000000  +
#####4##77####::::;<##????@@#CCC###FFFGGGGHHHHJKKLLNNNNO000000000000  +
#####4##77####::::;<##????@@#CCC###FFFGGGGHHHHJKKLLNNNNOPPPP0000000  +
#####4##77####::::;<##????@@#CCC###FFFGGGGHHHHJKK##NNNNOPPPP0000000  -
#####4##77####::::;<##????@@#CCC###FFFGGGGHHHHJKK##NNNNOPPPPRRR0000  +
#####4##77####::::;<##????@@#CCC###FFFGGGGHHHHJKK##NNNNOPPPPRRRS000  +
TT000400770000::::;<00????@@0CCC000FFFGGGGHHHHJKK00NNNNOPPPPRRRS000  +
##000400770000::::;<00????@@0CCC000FFFGGGGHHHHJKK00NNNNOPPPPRRRS000  -
##00040077VVVV::::;<00????@@0CCC000FFFGGGGHHHHJKK00NNNNOPPPPRRRS000  +
##WW040077VVVV::::;<00????@@0CCC000FFFGGGGHHHHJKK00NNNNOPPPPRRRS000  +
##WW0400##VVVV::::;<00????@@0CCC000FFFGGGGHHHHJKK00NNNNOPPPPRRRS000  -
##WWY400##VVVV::::;<00????@@0CCC000FFFGGGGHHHHJKK00NNNNOPPPPRRRS000  +
##WWY400##VVVV::::;<00????@@0###000FFFGGGGHHHHJKK00NNNNOPPPPRRRS000  -
##WWY400##VVVV::::#<00????@@0###000FFFGGGGHHHHJKK00NNNNOPPPPRRRS000  -
##WWY400##VVVV::::#<00????@@0###000FFFGGGGHHHHJ##00NNNNOPPPPRRRS000  -
##WWY4]0##VVVV::::#<00????@@0###000FFFGGGGHHHHJ##00NNNNOPPPPRRRS000  +
##WWY4]0##VVVV::::##00????@@0###000FFFGGGGHHHHJ##00NNNNOPPPPRRRS000  -
```

```
##WWY4]0##VVVV::::##00????@@0###000FFFGGGGHHHH###00NNNNOPPPPRRRS  -
##WW#4]0##VVVV::::##00????@@0###000FFFGGGGHHHH###00NNNNOPPPPRRRS  -
00WW04]000VVVV::::00aa????@@0000000FFFGGGGHHHH00000NNNNOPPPPRRRS  +
00WW04]000VVVV####00aa????@@0000000FFFGGGGHHHH00000NNNNOPPPPRRRS  -
00WW04]000########00aa????@@0000000FFFGGGGHHHH00000NNNNOPPPPRRRS  -
00WW04]000########00aa????##0000000FFFGGGGHHHH00000NNNNOPPPPRRRS  -
00WW04]000########00aa????##eeee000FFFGGGGHHHH00000NNNNOPPPPRRRS  +
00WW04]000########00aa????##eeee000FFFGGGGHHHH00000NNNNOPPPP###S  -
00WW04#000########00aa????##eeee000FFFGGGGHHHH00000NNNNOPPPP###S  -
00WW0##000########00aa????##eeee000FFFGGGGHHHH00000NNNNOPPPP###S  -
iiWW0##000########00aa????##eeee000FFFGGGGHHHH00000NNNNOPPPP###S  +
iiWW0##000########00aa????##eeee000FFFGGGG####00000NNNNOPPPP###S  -
iiWW0##kk0########00aa????##eeee000FFFGGGG####00000NNNNOPPPP###S  +
iiWW0##kk0########llaa????##eeee000FFFGGGG####00000NNNNOPPPP###S  +
iiWW0##kk0########llaa????##eeee000FFFGGGG####00000NNNNOPPPP####  -
iiWW0##kk0########llaa????##eeee000FFFGGGG####nnnn0NNNNOPPPP####  +
iiWW0##kk0########llaa????##eeeeoo0FFFGGGG####nnnn0NNNNOPPPP####  +
iiWWp##kkp########llaa????##eeeeoopFFFGGGG####nnnnpNNNNOPPPP####  +
iiWWp##kkp########llaa????##eeeeoopFFFGGGG####nnnnpNNNNO########  -
iiWWpr0kkp00000000llaa????00eeeeoopFFFGGGG0000nnnnpNNNNO00000000  +
iiWWprskkp00000000llaa????00eeeeoopFFFGGGG0000nnnnpNNNNO00000000  +
iiWWprskkpt0000000llaa????00eeeeoopFFFGGGG0000nnnnpNNNNO00000000  +
iiWWprskkpt0000000llaa????00eeeeoop###GGGG0000nnnnpNNNNO00000000  -
iiWWprskkptv000000llaa????00eeeeoop###GGGG0000nnnnpNNNNO00000000  +
iiWWprskkptvwww000llaa????00eeeeoop###GGGG0000nnnnpNNNNO00000000  +
iiWWprskkptvwwwx00llaa????00eeeeoop###GGGG0000nnnnpNNNNO00000000  +
iiWWprskkptvwwwx00llaa????00####oop###GGGG0000nnnnpNNNNO00000000  -
iiWWprskkptvwwwx00llaa????00####oop###GGGGzzzznnnnpNNNNO00000000  +
iiWWprskkptvwwwx00llaa????00####oop###GGGGzzzznnnnpNNNNO{{{{0000  +
```

Variant #3 - mälu vabastamisel märgistatakse vabastavad blokid prahina ja mälu reserveerimisel leitakse kõige sobivam blokk

```
22200000000000000000000000000000000000000000000000000000000000000  +
22233300000000000000000000000000000000000000000000000000000000000  +
22233344000000000000000000000000000000000000000000000000000000000  +
###33344000000000000000000000000000000000000000000000000000000000  -
###33344666600000000000000000000000000000000000000000000000000000  +
###33344666677700000000000000000000000000000000000000000000000000  +
###33344666677788800000000000000000000000000000000000000000000000  +
###333##666677788800000000000000000000000000000000000000000000000  -
###333##6666777888::000000000000000000000000000000000000000000000  +
###333##6666777888::;;;;;000000000000000000000000000000000000000  +
###333##6666777888::;;;;;<<<000000000000000000000000000000000000  +
###333##6666777888::;;;;;###000000000000000000000000000000000000  -
###333##6666777888::;;;;;###>>>000000000000000000000000000000000  +
###333##6666777888::;;;;;###>>>??000000000000000000000000000000  +
###333##6666777888::;;;;;###>>>??@@@@000000000000000000000000000  +
###333##6666777888##;;;;###>>>??@@@@000000000000000000000000000  -
###333##6666777888##;;;;###>>>??@@@@B00000000000000000000000000  +
###333##6666777888##;;;;###>>>??@@@@BCC0000000000000000000000  +
###333##6666777888##;;;;###>>>??@@@@BCCDDDD0000000000000000000  +
###333##6666777#####;;;;###>>>??@@@@BCCDDDD0000000000000000000  -
###333##6666777#####;;;;###>>>??@@@@BCCDDDDF000000000000000000  +
###333##6666777#####;;;;###>>>??@@@@BCCDDDDFGG00000000000000000  +
###333##6666777#####;;;;###>>>??@@@@BCCDDDDFGGHHH000000000000000  +
########6666777#####;;;;###>>>??@@@@BCCDDDDFGGHHH000000000000000  -
########6666777#####;;;;###>>>??@@@@BCCDDDDFGGHHHJ00000000000000  +
########6666777#####;;;;###>>>??@@@@BCCDDDDFGGHHHJKK0000000000000  +
```

```
########6666777#####;;;;###>>>??@@@@BCCDDDDFGGHHHJKKL00000000000  +
########6666777#####;;;;###>>>??@@@@BCCDDDDF##HHHJKKL00000000000  -
########6666777#####;;;;###>>>??@@@@BCCDDDDF##HHHJKKLNNN00000000  +
########6666777#####;;;;###>>>??@@@@BCCDDDDF##HHHJKKLNNNOOOO0000  +
########6666777#####;;;;###>>>??@@@@BCCDDDDF##HHHJKKLNNNOOOOPPP0  +
########6666777#####;;;;######??@@@@BCCDDDDF##HHHJKKLNNNOOOOPPP0  -
########6666777#####;;;;######??@@@@BCCDDDDF##HHHJKKLNNNOOOOPPPR  +
000000006666677700000;;;;000000??@@@@BCCDDDDFSSHHHJKKLNNNOOOOPPPR  +
000000006666777TTT00;;;;000000??@@@@BCCDDDDFSSHHHJKKLNNNOOOOPPPR  +
000000006666777TTT00;;;;000000??@@@@#CCDDDDFSSHHHJKKLNNNOOOOPPPR  -
000000006666777TTT00;;;;VVV000??@@@@#CCDDDDFSSHHHJKKLNNNOOOOPPPR  +
000000006666777TTT00;;;;VVVWWW??@@@@#CCDDDDFSSHHHJKKLNNNOOOOPPPR  +
000000006666777TTT00;;;;VVVWWW??@@@@#CCDDDDFSS###JKKLNNNOOOOPPPR  -
000000006666777TTTYY;;;;VVVWWW??@@@@#CCDDDDFSS###JKKLNNNOOOOPPPR  +
000000006666777TTTYY;;;;VVVWWW##@@@@#CCDDDDFSS###JKKLNNNOOOOPPPR  -
00000000####777TTTYY;;;;VVVWWW##@@@@#CCDDDDFSS###JKKLNNNOOOOPPPR  -
00000000####777TTT##;;;;VVVWWW##@@@@#CCDDDDFSS###JKKLNNNOOOOPPPR  -
]]]00000####777TTT##;;;;VVVWWW##@@@@#CCDDDDFSS###JKKLNNNOOOOPPPR  +
]]]00000####777TTT##;;;;VVVWWW##@@@@#CCDDDDFSS###JKK#NNNOOOOPPPR  -
###00000####777TTT##;;;;VVVWWW##@@@@#CCDDDDFSS###JKK#NNNOOOOPPPR  -
###00000####777TTT##;;;;VVVWWW##@@@@#CC####FSS###JKK#NNNOOOOPPPR  -
###aaaa0####777TTT##;;;;VVVWWW##@@@@#CC####FSS###JKK#NNNOOOOPPPR  +
###aaaa0####777TTT##;;;;VVVWWW##@@@@#CC#####SS###JKK#NNNOOOOPPPR  -
###aaaa0####777TTT##;;;;VVVWWW##@@@@########SS###JKK#NNNOOOOPPPR  -
###aaaa0####777TTT##;;;;###WWW##@@@@########SS###JKK#NNNOOOOPPPR  -
eeeaaaa00000777TTT00;;;;000WWW00@@@@00000000SS000JKK0NNNOOOOPPPR  +
eeeaaaa00000777TTT00;;;;000###00@@@@00000000SS000JKK0NNNOOOOPPPR  -
eeeaaaa00000777TTT00;;;;000###00@@@@00000000SS000JKK0NNN####PPPR  -
eeeaaaa00000###TTT00;;;;000###00@@@@00000000SS000JKK0NNN####PPPR  -
eeeaaaa00000###TTT00;;;;000###00@@@@00000000SS000J##0NNN####PPPR  -
eeeaaaajjjj0###TTT00;;;;000###00@@@@00000000SS000J##0NNN####PPPR  +
eeeaaaajjjj0###TTT00;;;;kkk###00@@@@00000000SS000J##0NNN####PPPR  +
eeeaaaajjjjl###TTT00;;;;kkk###00@@@@00000000SS000J##0NNN####PPPR  +
eeeaaaa####l###TTT00;;;;kkk###00@@@@00000000SS000J##0NNN####PPPR  -
eeeaaaa####l###TTTnn;;;;kkk###00@@@@00000000SS000J##0NNN####PPPR  +
eeeaaaa####l###TTTnn;;;;kkk###00@@@@00000000SS000J##oNNN####PPPR  +
eeeaaaa####l###TTTnn;;;;kkk###pp@@@@00000000SS000J##oNNN####PPPR  +
eeeaaaa####l###TTTnn;;;;kkk###pp@@@@00000000SS000J##o#######PPPR  -
eeeaaaa####l###TTTnn;;;;kkk###pp@@@@00000000SSrrrJ##o#######PPPR  +
eeeaaaa####l###TTTnn;;;;kkk###pp@@@@sss00000SSrrrJ##o#######PPPR  +
eeeaaaa####l###TTTnn;;;;kkk###pp@@@@sssttttt0SSrrrJ##o#######PPPR  +
eeeaaaa####l###TTTnn####kkk###pp@@@@sssttttt0SSrrrJ##o#######PPPR  -
eeeaaaa####l###TTTnn####kkk###pp@@@@ssstttttvSSrrrJ##o#######PPPR  +
eeeaaaa0000l000TTTnn0000kkk000pp@@@@ssstttttvSSrrrJwwo0000000PPPR  +
eeeaaaa0000lxxxTTTnn0000kkk000pp@@@@ssstttttvSSrrrJwwo0000000PPPR  +
###aaaa0000lxxxTTTnn0000kkk000pp@@@@ssstttttvSSrrrJwwo0000000PPPR  -
###aaaa0000lxxxTTTnn0000kkkzzzpp@@@@ssstttttvSSrrrJwwo0000000PPPR  +
###aaaa{{{0lxxxTTTnn0000kkkzzzpp@@@@ssstttttvSSrrrJwwo0000000PPPR  +
```

Kommentaarid

{Lugemise hõlbustamiseks nimetame algoritmid :

1. reserveerimisalgoritm - leitakse esimene sobiv auk
2. reserveerimisalgoritm - valitakse sobivaim auk

1. vabastamisalgoritm - vabastatavad blokid suunatakse kohe vabade blokkide hulka

2. vabastamisalgoritm - vabastatavad blokid märgistatakse prahina ja vajadusel praht koristatakse}

2. reserveerimisalgoritm on selles mõttes etem kui esimene, et ta teeb mälutükid kompaktsemaks ja vähendab mälu fragmenteerumist - et valitakse sobivaim (väikseim auk), siis jäävad suuremad augud suuremate mälublokkide tarvis. Samas on 1. reserveerimisalgoritm kiirem, kuna siin ei käida tingimata kogu mälu läbi augu otsinguil.

2. vabastamisalgoritmi eelis on see, et kui mälu on täis vaid informatsiooni ja prahti ja kui mälupuudusel praht ära koristatakse, siis vabaneb mälu korraga suurte tükkidena, mis võimaldab panna mällu suuremaid tükke neid fragmenteerimata. Samas jällegi, kui mälu pole nii täis, et prahikoristust oleks vaja teha, aga on nii täis, et suuremaid vabu blokke rohkem pole, siis fragmenteerub mälu jällegi rohkem. Selles suhtes on 1. vabastamisalgoritm parem, kuna kogu vaba mälu on alati kogu aeg kasutatav.

Konkreetselt öelda ei saa, milline algoritm kummastki paarist on parem, see sõltub olukorrast. Aga tundub, et kiirema arvuti ja mälu puhul on rentaabel kasutada 2. reserveerimisalgoritmi, kuna siis ei võta sobivaima bloki otsimine eriti rohkem aega kui esimese sobiva bloki otsimine.

_____-

Programmi lähtekood :

```
PROGRAM M2luJuhtimine;

uses crt;

const max = 64;
      maxtykk = 5;
      esimenesymbol = 49;

type m2lutyyp = string[max];

const tyhim2lu : m2lutyyp =
'0000000000000000000000000000000000000000000000000000000000000000';

type paar = record
      koht : m2lutyyp;
     end;

     mass = array[1..max] of paar;

var aitabkah : boolean;
    m2lu, abim2lu : m2lutyyp;
    miskus : mass;
    i, j, k : byte;
    hetksuurus, hetkkoht, tykkidearv, midateha, v2ikeloendur : byte;
    fail : text;
    variant1, variant2 : char;
```

```
FUNCTION Xorstring (xoritav1, xoritav2 : m2lutyyp) : m2lutyyp;
var ajutinexor : m2lutyyp;

begin
 ajutinexor := xoritav1;
 if variant2 = '1' then
  begin
  for i := 1 to max do
   if xoritav2[i] <> '0' then if xoritav1[i] <> '0' then ajutinexor[i] := '0'
                              else ajutinexor[i] := xoritav2[i];
  end
 else for i := 1 to max do
  if not (xoritav2[i] in ['0', '#']) then if not (xoritav1[i] in ['0', '#'])
                                          then ajutinexor[i] := '#'
                                          else ajutinexor[i] := xoritav2[i];
  xorstring := ajutinexor;
end;



FUNCTION T2idetud (t2ism2lu : m2lutyyp) : byte;
var t2isabi : byte;

begin
     t2isabi := 0;
     for i := 1 to max do if not (t2ism2lu[i] in ['0', '#']) then inc(t2isabi);
     T2idetud := t2isabi;
end;



FUNCTION Leiakohad1 (leitavsuurus : byte) : m2lutyyp;
var aukk2es, leitud : boolean;
    leitudsuurus : byte;
    abikohad : m2lutyyp;
begin
 leiakohad1 := tyhim2lu;
 aukk2es := false;
 for i := 1 to max - leitavsuurus + 1 do
  begin
  leitud := true;
  for j := i to i + leitavsuurus - 1 do if m2lu[j] <> '0' then leitud := false;
   if leitud then
   begin
   for j := i to i + leitavsuurus - 1 do leiakohad1[j] := chr(esimenesymbol +
midateha);
   exit;
   end;
  end;
 k := leitavsuurus - 1;
 leitudsuurus := 0;
 abikohad := m2lu;
 if (leitavsuurus > 1) then
  repeat
    i := 0;
    repeat
    inc (i);
    leitud := true;
    for j := i to i + k - 1 do if abikohad[j] <> '0' then leitud := false;
    if leitud then
     begin
     for j := i to i + k - 1 do leiakohad1[j] := chr(esimenesymbol + midateha);
     for j := i to i + k - 1 do abikohad[j] := chr(esimenesymbol + midateha);
     leitudsuurus := leitudsuurus + k;
     end;
    until (leitud = true) or (i = max - k + 1);
  if (k = 1) and not leitud then
```

```
   begin
   leiakohad1 := tyhim2lu;
   aukk2es := true;
   end;
  if not leitud then k := k - 1;
  if leitudsuurus = leitavsuurus then aukk2es := true;
  until aukk2es;
end;



FUNCTION Leiakohad2 (leitavsuurus : byte) : m2lutyyp;
var aukk2es, leitud : boolean;
    leitudsuurus : integer;
    abikohad, teineabi : m2lutyyp;
    aukmass : array[1..max] of integer;
    abi, l, paljuleitud : byte;

begin
 for i := 1 to max do aukmass[i] := 0;
 leiakohad2 := tyhim2lu;
 aukk2es := false;
 abi := 0;
 i := 0;
 leitudsuurus := 0;
 repeat
 inc (i);
 if m2lu[i] = '0' then
  begin
  k := 0;
  j := i - 1;
   repeat
   inc (j);
   if m2lu[j] = '0' then inc(k);
   until not (m2lu[j] = '0');
  inc (abi);
  aukmass[abi] := i*100 + k;
  if k > leitudsuurus mod 100 then leitudsuurus := i*100 + k;
  i := i + k - 1;
 end;
 until i > max - leitavsuurus;
 for i := 1 to abi do
  begin
  if (aukmass[i] mod 100 > (leitavsuurus mod 100 - 1)) and (aukmass[i] mod 100 <
leitudsuurus mod 100) then
   leitudsuurus := aukmass[i]
  end;
 if leitudsuurus mod 100 >= leitavsuurus then
  begin
  for i := leitudsuurus div 100 to leitudsuurus div 100 + leitavsuurus - 1 do
   leiakohad2[i] := chr(esimenesymbol + midateha);
  aukk2es := true;
  end;




 if (leitavsuurus = 1) or aukk2es then exit;
 l := leitavsuurus - 1;
 abi := 0;
 paljuleitud := 0;
 leitud := true;
 teineabi := tyhim2lu;
 abikohad := m2lu;
 for i := 1 to max do aukmass[i] := 0;
  repeat
    if leitud then
     begin
     leitud := false;
     abi := 0;
```

```
     leitudsuurus := 0;
     i := 0;
     repeat
     inc (i);
     if abikohad[i] = '0' then
      begin
      k := 0;
      j := i - 1;
       repeat
       inc (j);
       if abikohad[j] = '0' then inc(k);
       until not (abikohad[j] = '0');
      inc (abi);
      aukmass[abi] := i*100 + k;
      if k > leitudsuurus mod 100 then leitudsuurus := i*100 + k;
      i := i + k - 1;
      end;
     until i > max - l;
     end;
    for i := 1 to abi do
     begin
     if (aukmass[i] mod 100 > (l - 1)) and (aukmass[i] mod 100 < leitudsuurus
mod 100) then
      leitudsuurus := aukmass[i]
     end;
    if leitudsuurus mod 100 >= l then
     begin
     for i := leitudsuurus div 100 to leitudsuurus div 100 + l - 1 do
      begin
      teineabi[i] := chr(esimenesymbol + midateha);
      abikohad[i] := chr(esimenesymbol + midateha);
      end;
     leitud := true;
     paljuleitud := paljuleitud + l;
    end;



  if (l = 1) and not leitud then
   begin
   teineabi := tyhim2lu;
   aukk2es := true;
   end;
  if not leitud then l := l - 1;
  if paljuleitud + l > leitavsuurus then dec (l);
  if paljuleitud = leitavsuurus then aukk2es := true;
  until aukk2es;
  leiakohad2 := teineabi;
end;




FUNCTION M2luvabaks (vabaksmitmes : byte) : m2lutyyp;

begin
 m2luvabaks := xorstring (m2lu, miskus[vabaksmitmes].koht);
 for i := vabaksmitmes to tykkidearv - 1 do miskus[i] := miskus[i + 1];
 dec (tykkidearv);
end;



PROCEDURE Prahtminema;

begin
 for i := 1 to max do if m2lu[i] = '#' then m2lu[i] := '0';
end;
```

```
BEGIN
Randomize;
 m2lu := tyhim2lu;
 tykkidearv := 0;
 midateha := 0;
 Assign (fail, 'mudel.txt');
 Rewrite (fail);
 writeln ('M„lu reserveerimisel leitakse');
 Writeln ('1 - esimene sobiv blokk');
 Writeln ('2 - sobivaim blokk');
 repeat variant1 := readkey until variant1 in ['1'..'2'];

 writeln;
 writeln ('M„lu vabastamisel vabastatavad blokid');
 writeln ('1 - suunatakse kohe vabade blokkide hulka');
 writeln ('2 - m„rgistatakse prahina ja koristatakse vajadusel');
 repeat variant2 := readkey until variant2 in ['1'..'2'];

 write (fail, 'M„lu reserveerimisel leitakse ');
 if variant1 = '1' then writeln (fail, 'esimene sobiv blokk.')
                   else writeln (fail, 'sobivaim blokk.');
 write (fail, 'M„lu vabastamisel vabastatavad blokid ');
 if variant2 = '1' then writeln (fail, 'suunatakse kohe vabade blokkide hulka.')
                   else writeln (fail, 'm„rgistatakse prahina.');
 writeln (fail);


 repeat
 inc (midateha);
 if midateha mod 4 = 0 then
  begin
  hetksuurus := random (tykkidearv) + 1;
  m2lu := m2luvabaks (hetksuurus);
  end
 else
  begin
  v2ikeloendur := 0;
  if variant1 = '2' then
   repeat
   inc (v2ikeloendur);
   hetksuurus := random (maxtykk - 1) + 1;
   abim2lu := Leiakohad2 (hetksuurus);
   if v2ikeloendur > 1 then prahtminema;
   until t2idetud(abim2lu) > 0
  else
   begin
   if variant2 = '2' then
    repeat
    inc (v2ikeloendur);
    hetksuurus := random (maxtykk - 1) + 1;
    abim2lu := Leiakohad1 (hetksuurus);
    if v2ikeloendur > 1 then prahtminema;
    until t2idetud(abim2lu) > 0
   else
    begin
    hetksuurus := random (maxtykk - 1) + 1;
    abim2lu := leiakohad1 (hetksuurus);
    end;
   end;
  inc (tykkidearv);
  miskus [tykkidearv].koht := abim2lu;
  m2lu := xorstring (m2lu, abim2lu);
  end;



 write (fail, m2lu);
 write (fail, '  ');
 if midateha mod 4 = 0 then writeln (fail, '-') else writeln (fail, '+');
 until t2idetud(m2lu) > 50;

 repeat
```

```
inc (midateha);
if midateha mod 4 = 0 then
 begin
 v2ikeloendur := 0;
 if variant1 = '2' then
  repeat
  inc (v2ikeloendur);
  hetksuurus := random (maxtykk - 1) + 1;
  abim2lu := Leiakohad2 (hetksuurus);
  if v2ikeloendur > 1 then prahtminema;
  until t2idetud(abim2lu) > 0
 else
  begin
  if variant2 = '2' then
   repeat
   inc (v2ikeloendur);
   hetksuurus := random (maxtykk - 1) + 1;
   abim2lu := Leiakohad1 (hetksuurus);
   if v2ikeloendur > 1 then prahtminema;
   until t2idetud(abim2lu) > 0
  else
   begin
   hetksuurus := random (maxtykk - 1) + 1;
   abim2lu := leiakohad1 (hetksuurus);
   end;
  end;
 inc (tykkidearv);
 miskus [tykkidearv].koht := abim2lu;
 m2lu := xorstring (m2lu, abim2lu);
 end
else
 begin
 hetksuurus := random (tykkidearv) + 1;
 m2lu := m2luvabaks (hetksuurus);
 end;



write (fail, m2lu);
write (fail, '  ');
if midateha mod 4 = 0 then writeln (fail, '+') else writeln (fail, '-');
until t2idetud(m2lu) < 32;

repeat
inc (midateha);
if midateha mod 4 = 0 then
 begin
 hetksuurus := random (tykkidearv) + 1;
 m2lu := m2luvabaks (hetksuurus);
 end
else
 begin
 v2ikeloendur := 0;
 if variant1 = '2' then
  repeat
  inc (v2ikeloendur);
  hetksuurus := random (maxtykk - 1) + 1;
  abim2lu := Leiakohad2 (hetksuurus);
  if v2ikeloendur > 1 then prahtminema;
  until t2idetud(abim2lu) > 0
 else
  begin
  if variant2 = '2' then
   repeat
   inc (v2ikeloendur);
   hetksuurus := random (maxtykk - 1) + 1;
   abim2lu := Leiakohad1 (hetksuurus);
   if v2ikeloendur > 1 then prahtminema;
   until t2idetud(abim2lu) > 0
  else
   begin
   hetksuurus := random (maxtykk - 1) + 1;
```

```
    abim2lu := leiakohad1 (hetksuurus);
    end;
   end;
  inc (tykkidearv);
  miskus [tykkidearv].koht := abim2lu;
  m2lu := xorstring (m2lu, abim2lu);
  end;



 write (fail, m2lu);
 write (fail, '  ');
 if midateha mod 4 = 0 then writeln (fail, '-') else writeln (fail, '+');
 until t2idetud(m2lu) > 47;
 writeln (fail, 'M„luoperatsioone sooritati ', midateha, ' korda.');
 Close (fail);
 writeln (midateha);
END.
```