Ü»ñ³ÍáõÃÛáõÝ

Ä³Ù³Ý³Ï³ÏÇó Íñ³·ñ³ÛÇÝ Ñ³Ù³Ï³ñ·»ñÁ Ý³Ë³·ÍíáõÙ »Ý Ùá¹áõÉ³ÛÇÝ ëÏ½µáõÝùáí: Øá¹áõÉÇ Çñ³Ï³Ý³óÙ³Ý »Õ³Ý³ÏÁ ëáíáñ³µ³ñ Ã³ùÝí³Í ¿ û·ï³·áñÍáÕÇó, ÇëÏ Ñ³Ù³·áñÍ³ÏóáõÃÛáõÝÁ Ùá¹áõÉÇ Ñ»ï Ï³ï³ñíáõÙ ¿ ÇÝï»ñý»ÛëÇ` Ùá¹áõÉÇÝ ÷áË³ÝóíáÕ ¨ Ùá¹áõÉÇó ëï³óíáÕ å³ñ³Ù»ïñ»ñÇ ÙÇçáóáí: ²Ûë Ùáï»óáõÙÁ Ñ³Ù³ñíáõÙ ¿ ý‎‎‎‎‎‎‎‎‎áõÝÏóÇáÝ³É ³µëïñ³ÏóÇ³ÛÇ Çñ³Ï³Ý³óÙ³Ý »Õ³Ý³Ï: Ìñ³·ñ³ÛÇÝ Ñ³Ù³Ï³ñ·»ñÁ Ý³Ë³·Í»ÉÇë ³éÏ³ ¿ Ý³¨ ïíÛ³ÉÝ»ñÇ ³µëïñ³ÏóÇ³Ý: Ìñ³·ñáñ¹ÇÝ ëáíáñ³µ³ñ Ñ³ÛïÝÇ »Ý ÙÇ³ÛÝ ³ÛÝ ·áñÍáÕáõÃÛáõÝÝ»ñÁ, áñáÝù Ï³ñ»ÉÇ ¿ Ï³ï³ñ»É ³Û¹ ïíÛ³ÉÝ»ñÇ ÝÏ³ïÙ³Ùµ: ú·ïíáÕÇÝ áã ÙÇßï »Ý Ñ³ë³Ý»ÉÇ ÇÝãå»ë ïíÛ³ÉÝ»ñÇ Ý»ñÏ³Û³óÙ³Ý, ³ÛÝå»ë ¿É ·áñÍáÕáõÃÛáõÝÝ»ñÇ Çñ³Ï³Ý³óÙ³Ý Ó¨»ñÁ: îíÛ³ÉÝ»ñÇ Ï³½Ù³Ï»ñåÙ³Ý ³Û¹ ëÏ½µáõÝùÇ Ñ»ï Ï³åí³Í ¿ ïíÛ³ÉÝ»ñÇ ³µëïñ³Ïï ïÇå Ñ³ëÏ³óáõÃÛáõÝÁ (Abstract Data Type, ADT): ADT ³ë»Éáí Ñ³ëÏ³ÝáõÙ »Ýù ïíÛ³ÉÝ»ñÇ Ù³Ã»Ù³ïÇÏ³Ï³Ý Ùá¹»É ³ÛÝ ·áñÍáÕáõÃÛáõÝÝ»ñÇ Ñ³Ù³ËÙµáí, áñáÝù Ï³ï³ñíáõÙ »Ý Ùá¹»ÉáõÙ ÁÝ¹·ñÏí³Í ïíÛ³ÉÝ»ñÇ ÝÏ³ïÙ³Ùµ:

úñÇÝ³Ï, ïíÛ³ÉÝ»ñÇ ³µëïñ³Ïï ïÇå»ñ »Ý Ñ³Ý¹Çë³ÝáõÙ.

1. ¶Í³ÛÇÝ óáõó³ÏÁ, áñÇ Ñ»ï Ï³ï³ñíáÕ ·áñÍáÕáõÃÛáõÝÝ»ñÝ »Ý ï³ññÇ ³í»É³óáõÙÁ, ï³ññÇ Ñ»é³óáõÙÁ, óáõó³ÏÝ»ñÇ ÏóáõÙÁ ¨ ³ÛÉÝ:

2. àñáß³ÏÇ ïÇåÇ ï³ññ»ñÇ µ³½ÙáõÃÛáõÝÁ` ï³ññÇ å³ïÏ³Ý»ÉáõÃÛáõÝ ëïáõ·Ù³Ý, µ³½ÙáõÃÛáõÝÝ»ñÇ ÙÇ³íáñÙ³Ý, Ñ³ïÙ³Ý ¨ ³ÛÉ ·áñÍáÕáõÃÛáõÝÝ»ñÇ Ñ³Ù³ËÙµáí:

3. Ì³éÁ ` Ñ³Ý·áõÛóÝ»ñÇ Ùß³ÏÙ³Ý ·áñÍáÕáõÃÛáõÝÝ»ñÇ Ñ³Ù³ËÙµáí:

²µëïñ³Ïï ïÇåÇ ïíÛ³ÉÝ»ñÇ Ý»ñÏ³Û³óÙ³Ý Ó¨»ñÁ, áñáÝù ³ç³ÏóíáõÙ »Ý Íñ³·ñ³íáñÙ³Ý É»½íÇ ÙÇçáóÝ»ñáí, ³Ýí³ÝáõÙ »Ý ïíÛ³ÉÝ»ñÇ Ï³éáõóí³ÍùÝ»ñ: îíÛ³ÉÝ»ñÇ Ï³éáõóí³Íùáí áñáßíáõÙ ¿ ADT-Ç ·áñÍáÕáõÃÛáõÝÝ»ñÇ Çñ³Ï³Ý³óáõÙÁ:

1. îíÛ³ÉÝ»ñÇ ³µëïñ³Ïï ïÇå»ñ

îíÛ³ÉÝ»ñÇ ADT ³µëïñ³Ïï ïÇåÁ ¹Çï³ñÏ»Ýù áñå»ë Ù³Ã»Ù³ïÇÏ³Ï³Ý Ùá¹»É Çñ ·áñÍáÕáõÃÛáõÝÝ»ñÇ µ³½ÙáõÃÛ³Ùµ, áñáÝù áñáßí³Í »Ý ³Û¹ Ùá¹»ÉÇ Ù»ç: ADT-Ý ÏáÝÏñ»ï É»½íáõÙ Çñ³Ï³Ý³óÝ»ÉÇë û·ï³·áñÍíáõÙ ¿ ïíÛ³ÉÝ»ñÇ Ï³éáõóí³ÍùÝ»ñ:

Ìñ³·ñÇ Ý³Ë³·ÍÙ³Ý ¨ Ùß³ÏÙ³Ý Ñ³Ù³ñ Ï³ñ¨áñ ¿ ïíÛ³ÉÝ»ñÇ ³µëïñ³ÏóÇ³Ý: ²µëïñ³ÏóÇ³Ý áñáßáõÙ ¿ ïíÛ³ÉÝ»ñÇ Ï³éáõóí³ÍùÁ ¨ ïíÛ³ÉÝ»ñÇ Ñ»ï Ï³ï³ñíáÕ ·áñÍáÕáõÃÛáõÝÝ»ñÁ: ²µëïñ³Ïï ïÇåÇ ïíÛ³ÉÝ»ñÁ Ï³½ÙáõÙ »Ý û·ïíáÕÇ ÏáÕÙÇó ë³ÑÙ³ÝíáÕ ïÇå, áñÇ ·áñÍáÕáõÃÛáõÝÝ»ñÁ óáõÛó »Ý ï³ÉÇë, Ã» û·ïíáÕÝ ÇÝãå»ë Ï³ñáÕ ¿ ³ßË³ï»É ³Û¹ ïíÛ³ÉÝ»ñÇ Ñ»ï:

ADT-Ý ÝÏ³ñ³·ñíáõÙ ¿ áñáß³ÏÇ ýáñÙ³ïáí, áñÁ å³ñáõÝ³ÏáõÙ ¿ í»ñÝ³·Çñ, ïíÛ³ÉÝ»ñÇ ïÇåÇ ÝÏ³ñ³·Çñ, ·áñÍáÕáõÃÛáõÝÝ»ñÇ óáõó³Ï:

Úáõñ³ù³ÝãÛáõñ ·áñÍáÕáõÃÛ³Ý Ñ³Ù³ñ áñáßíáõÙ »Ý`

* Øáõïù³ÛÇÝ ³ñÅ»ùÝ»ñ (input ),
* Ü³Ë³å³ÛÙ³ÝÝ»ñ (preconditions), áñáÝó µ³í³ñ³ñáõÙ »Ý ïíÛ³ÉÝ»ñÁ Ý³Ëù³Ý Ýñ³Ýó Ñ»ï ·áñÍáÕáõÃÛáõÝ Ï³ï³ñ»ÉÁ,
* äñáó»ë (process), áñÁ ÝÏ³ñ³·ñáõÙ ¿ Ï³ï³ñíáÕ ·áñÍáÕáõÃÛáõÝÁ,
* ºÉù³ÛÇÝ ³ñÅ»ùÝ»ñ (output),
* ì»ñçÝ³å³ÛÙ³Ý (postconditions), áñáÝó µ³í³ñ³ñáõÙ »Ý ïíÛ³ÉÝ»ñÁ ·áñÍáÕáõÃÛáõÝÁ Ï³ï³ñ»Éáõó Ñ»ïá:

ADT-Ý»ñÇ Ù»Í Ù³ëÝ áõÝ»Ý ëÏ½µÝ³ñÅ»ù³íáñÙ³Ý ·áñÍáÕáõÃÛáõÝÝ»ñ (initializer),áñáÝù ïíÛ³ÉÝ»ñÇÝ ï³ÉÇë »Ý ëÏ½µÝ³Ï³Ý ³ñÅ»ùÝ»ñ: C++ É»½íáõÙ ïíÛ³ÉÝ»ñÇ ëÏ½µÝ³ñÅ»ù³íáñáõÙÁ Çñ³Ï³Ý³óíáõÙ ¿ ¹³ëÇ ÏáÝëïñáõÏïáñÇ (constructor) ÙÇçáóáí: ADT-Ç ÝÏ³ñ³·ñáõÃÛáõÝÁ ïñíáõÙ ¿ Ñ»ï¨Û³É Ó¨áí:

**ADT** ADT-³ÝáõÝ

*îíÛ³ÉÝ»ñ*

îíÛ³ÉÝ»ñÇ Ï³éáõóí³ÍùÇ ÝÏ³ñ³·Çñ:

*¶áñÍáÕáõÃÛáõÝÝ»ñ*

* ÎáÝëïñáõÏïáñ

êÏ½µÝ³Ï³Ý ³ñÅ»ùÝ»ñ - ïíÛ³ÉÝ»ñ, áñáÝù û·ï³·áñÍíáõÙ »Ý ûµÛ»ÏïÁ ëÏ½µÝ³ñÅ»ù³íáñ»ÉÇë

äñáó»ë - ûµÛ»ÏïÇ ëÏ½µÝ³ñÅ»ù³íáñáõÙ

* ¶áñÍáÕáõÃÛáõÝ - 1

Øáõïù – Ùáõïù³ÛÇÝ ïíÛ³ÉÝ»ñ

Ü³Ë³å³ÛÙ³Ý - ·áñÍáÕáõÃÛ³Ý Çñ³Ï³Ý³óÙ³Ý Ñ³Ù³ñ ³ÝÑñ³Å»ßï å³ÛÙ³ÝÝ»ñÇ Ñ³í³ù³Íáõ

äñáó»ë – ïíÛ³ÉÝ»ñÇ Ñ»ï Ï³ï³ñíáÕ ·áñÍáÕáõÃÛáõÝ

ºÉù – í»ñ³¹³ñÓíáÕ ïíÛ³ÉÝ»ñ

ì»ñçÝ³å³ÛÙ³Ý – å³ÛÙ³ÝÝ»ñÇ Ñ³í³ù³Íáõ,áñáÝù ï»ÕÇ áõÝ»Ý ·áñÍáÕáõÃÛáõÝÁ Ï³ï³ñ»Éáõó Ñ»ïá

* ¶áñÍáÕáõÃÛáõÝ - 2

…………………….

* ¶áñÍáÕáõÃÛáõÝ - n

**ì»ñç ADT**-³ÝáõÝ

1. è³óÇáÝ³É ÃÇíÁ Ý»ñÏ³Û³óíáõÙ ¿ ³ÝÏñ×³ï»ÉÇ Ïáïáñ³ÏÇ ï»ëùáí:
2. Øß³Ï»ù ¨ Çñ³Ï³Ý³óñ»ù ADT` Çñ³Ï³Ý ÃÇíÁ ýÇùë³Í ëïáñ³Ï»ïÇ ýáñÙ³ïáí Ý»ñÏ³Û³óÝ»Éáõ Ñ³Ù³ñ: Ü³Ë³ï»ë»ù Ãí»ñÇ ·áõÙ³ñÙ³Ý , Ñ³ÝÙ³Ý Ñ³Ù»Ù³ïÙ³Ý ·áñÍáÕáõÃÛáõÝÝ»ñÁ:
3. Øß³Ï»ù ¨ Çñ³Ï³Ý³óñ»ù ADT` ÏáÙåÉ»ùë ÃÇí Ý»ñÏ³Û³óÝ»Éáõ Ñ³Ù³ñ: Ü³Ë³ï»ë»ù ÏáÙåÉ»ùë Ãí»ñÇ ·áõÙ³ñÙ³Ý, µ³½Ù³å³ïÏÙ³Ý ¨ ³ñï³ÍÙ³Ý ·áñÍáÕáõÃÛáõÝÝ»ñÁ:

2.¶Í³ÛÇÝ óáõó³Ï

2.1 ¶Í³ÛÇÝ óáõó³Ï ïíÛ³ÉÝ»ñÇ ³µëïñ³Ïï ïÇåÁ

¶Í³ÛÇÝ óáõó³Ï ¿ ÏáãíáõÙ ÙÇ³ïÇå ï³ññ»ñÇ a1, . . . . , an (n >= 0) Ñ³çáñ¹³Ï³ÝáõÃÛáõÝÁ, áñÇ Ñ³Ù³ñ ë³ÑÙ³Ýí³Í »Ý Ý³Ëáñ¹ ï³ññ ¨ Ñ³çáñ¹ ï³ññ Ñ³ëÏ³óáõÃÛáõÝÝ»ñÁ Ñ»ï¨Û³É Ï»ñå` ai –Ç Ñ³Ù³ñ ai+1 – Á Ñ³çáñ¹ ï³ññÝ ¿ (1<=i<n), ÇëÏ ai-1–Á` Ý³Ëáñ¹(1<i<=n): ¶Í³ÛÇÝ óáõó³ÏÁ Ï³ñáÕ ¿ Ý³¨ ï³ññ»ñ ãå³ñáõÝ³Ï»É (n= 0): ²Û¹åÇëÇ óáõó³ÏÁ ÏáãíáõÙ ¿ ¹³ï³ñÏ óáõó³Ï:

¶Í³ÛÇÝ óáõó³Ï ADT-Ç ÝÏ³ñ³·ñáõÃÛáõÝÁ Ñ»ï¨Û³ÉÝ ¿`

ADT List

*îíÛ³ÉÝ»ñ*

¶Í³ÛÇÝ óáõó³Ï

*¶áñÍáÕáõÃÛáõÝÝ»ñ*

ÎáÝëïñáõÏïáñ

Ùáõïù – ãÏ³

Ý³Ë³å³ÛÙ³Ý – ãÏ³

åñáó»ë - ¹³ï³ñÏ óáõó³ÏÇ ëÏ½µÝ³ñÅ»ù³íáñáõÙ

»Éù –ãÏ³

í»ñçÝ³å³ÛÙ³Ý – ¹³ï³ñÏ óáõó³Ï

Length

Ùáõïù – ãÏ³

Ý³Ë³å³ÛÙ³Ý – ãÏ³

åñáó»ë – óáõó³ÏÇ ï³ññ»ñÇ ù³Ý³ÏÇ áñáßáõÙ

»Éù – óáõó³ÏÇ ï³ññ»ñÇ ù³Ý³Ï

í»ñçÝ³å³ÛÙ³Ý – ãÏ³

Empty

Ùáõïù – ãÏ³

Ý³Ë³å³ÛÙ³Ý – ãÏ³

åñáó»ë – óáõó³ÏÇ ¹³ï³ñÏ ÉÇÝ»Éáõ å³ÛÙ³ÝÇ ëïáõ·áõÙ

»Éù – 1(true), »Ã» óáõó³ÏÁ ¹³ï³ñÏ ¿, 0 (false) Ñ³Ï³é³Ï ¹»åùáõÙ:

í»ñçÝ³å³ÛÙ³Ý – ãÏ³

Find

Ùáõïù - óáõó³ÏáõÙ ÷ÝïñíáÕ ï³ññÁ

Ý³Ë³å³ÛÙ³Ý – ãÏ³

åñáó»ë – ï³ññÇ ÷ÝïñáõÙ óáõó³ÏáõÙ

»Éù – ÷ÝïñíáÕ ï³ññÇ ¹ÇñùÁ óáõó³ÏáõÙ

í»ñçÝ³å³ÛÙ³Ý – ãÏ³

Insert

Ùáõïù - óáõó³ÏÇÝ ³í»É³óíáÕ ï³ññÁ ¨ Ýñ³ ¹ÇñùÁ

Ý³Ë³å³ÛÙ³Ý – ãÏ³

åñáó»ë – ï³ññÇ ³í»É³óáõÙ óáõó³ÏáõÙ ïñí³Í ¹ÇñùáõÙ

»Éù – ãÏ³

í»ñçÝ³å³ÛÙ³Ý – ³í»É³óí³Í ï³ññáí óáõó³Ï

µ³ó³éÇÏ Çñ³íÇ×³Ï –- ³í»É³óíáÕ ï³ññÇ ¹ÇñùÁ ÃáõÛÉ³ïñ»ÉÇ ë³ÑÙ³ÝÝ»ñáõÙ ã¿

Delete

Ùáõïù – Ñ»é³óíáÕ ï³ññÇ ¹ÇñùÁ

Ý³Ë³å³ÛÙ³Ý – óáõó³ÏÁ ¹³ï³ñÏ ã¿

åñáó»ë – ï³ññÇ Ñ»é³óáõÙ óáõó³ÏÇó

»Éù – Ñ»é³óí³Í ï³ññÁ

í»ñçÝ³å³ÛÙ³Ý - Ñ»é³óí³Í ï³ññáí óáõó³Ï

µ³ó³éÇÏ Çñ³íÇ×³Ï –- Ñ»é³óíáÕ ï³ññÇ ¹ÇñùÁ ÃáõÛÉ³ïñ»ÉÇ ë³ÑÙ³ÝÝ»ñáõÙ ã¿

ClearList

Ùáõïù – ãÏ³

Ý³Ë³å³ÛÙ³Ý - ãÏ³

åñáó»ë – óáõó³ÏÇ µáÉáñ ï³ññ»ñÇ Ñ»é³óáõÙ

»Éù – ãÏ³

í»ñçÝ³å³ÛÙ³Ý –¹³ï³ñÏ óáõó³Ï

í»ñç ADT List

2.2¶Í³ÛÇÝ óáõó³ÏÇ Çñ³Ï³Ý³óáõÙ

¶Í³ÛÇÝ óáõó³ÏÁ Çñ³Ï³Ý³óíáõÙ ¿ ïíÛ³ÉÝ»ñÇ ³ÛÝåÇëÇ Ï³éáõóí³ÍùÝ»ñÇ ÙÇçáóáí, áñáÝù ÑÇÙÝí³Í »Ý ÑÇßáÕáõÃÛáõÝáõÙ ïíÛ³ÉÝ»ñÇ å³Ñå³ÝÙ³Ý Ñ³çáñ¹³Ï³Ý Ï³Ù Ï³å³Ïóí³Í »Õ³Ý³ÏÝ»ñÇ íñ³:

îíÛ³ÉÝ»ñÇ å³Ñå³ÝÙ³Ý Ñ³çáñ¹³Ï³Ý »Õ³Ý³ÏÇ ¹»åùáõÙ óáõó³ÏÇ Çñ³Ï³Ý³óáõÙÁ Ï³ï³ñíáõÙ ¿ ½³Ý·í³ÍÇ ÙÇçáóáí:

îíÛ³ÉÝ»ñÇ å³Ñå³ÝÙ³Ý Ï³å³Ïóí³Í »Õ³Ý³ÏÇ ¹»åùáõÙ ·Í³ÛÇÝ óáõó³ÏÇ Çñ³Ï³Ý³óáõÙÁ Ï³ï³ñíáõÙ ¿ Ï³å³Ïóí³Í óáõó³ÏÇ ÙÇçáóáí:

Î³å³Ïóí³Í óáõó³ÏÇ Ñ³Ý·áõÛóÁ áõÝÇ Ñ»ï¨Û³É Ï³éáõóí³ÍùÁ`

|  |  |
| --- | --- |
| INFO | LINK |

áñï»Õ INFO-Ý ÇÝýáñÙ³óÇáÝ ¹³ßïÝ ¾, ÇëÏ LINK-Á` Ñ³çáñ¹ Ñ³Ý·áõÛóÇ Ñ»ï Ï³åÇ ¹³ßïÁ:

ºÃ» x-Á óáõóÇã ¿ Ñ³Ý·áõÛóÇ íñ³, ³å³ INFO(x)-áí Ýß³Ý³Ï»Ýù ³Û¹ Ñ³Ý·áõÛóÇ ÇÝýáñÙ³óÇáÝ ¹³ßïÁ, ÇëÏ LINK(x) –áí` Ï³åÇ ¹³ßïÁ: ºÝÃ³¹ñíáõÙ ¿, áñ óáõó³ÏÇ ï³ññ»ñÁ ï»Õ³Ï³Ûí³Í »Ý ¹ÇÝ³ÙÇÏ Ó¨áí µ³ßËíáÕ ÑÇßáÕáõÃÛ³Ý Ù»ç` ÏáõÛïáõÙ, áñÁ ÝáõÛÝå»ë Çñ»ÝÇó Ý»ñÏ³Û³óÝáõÙ ¿ ÝáõÛÝ ïÇåÇ Ñ³Ý·áõÛóÝ»ñÇó Ï³½Ùí³Í Ï³å³Ïóí³Í óáõó³Ï: AVAIL-áí Ýß³Ý³Ï»Ýù óáõóÇãÁ ÏáõÛïÇ ³é³çÇÝ ï³ññÇ íñ³:

²é³ÝÓÝ³óÝ»Ýù óáõó³ÏÝ»ñÇ Ñ»ï Ï³åí³Í »ñÏáõ íÇ×³ÏÝ»ñ` OVERFLOW ¨ UNDERFLOW: OVERFLOW (·»ñÑ³·»óáõÙ) Ï³Ýí³Ý»Ýù ³ÛÝ Çñ³íÇ×³ÏÁ, »ñµ óáõó³ÏÝ ³ñ¹»Ý Éóí³Í ¿ ¨ ÷áñÓ ¿ ³ñíáõÙ óáõó³ÏÇÝ ³í»É³óÝ»É Ýáñ ï³ññ: UNDERFLOW-Ý (å³Ï³ëáõÙ) ³ÛÝ Çñ³íÇ×³ÏÝ ¿, »ñµ ÷áñÓ ¿ ³ñíáõÙ ¹³ï³ñÏ óáõó³ÏÇó Ñ»é³óÝ»É ï³ññ: ²Ûë ¹»åù»ñáõÙ Íñ³·ÇñÁ Ñ³Ù³å³ï³ëË³Ý Ó¨áí å»ïù ¿ ³ñÓ³·³ÝùÇ ëï»ÕÍí³Í Çñ³íÇ×³ÏÇÝ:

¸Çï³ñÏ»Ýù ÏáõÛïÇó Ñ³Ý·áõÛó í»ñóÝ»Éáõ ¨ Ñ³Ý·áõÛóÁ ÏáõÛï í»ñ³¹³ñÓÝ»Éáõ ·áñÍáÕáõÃÛáõÝÝ»ñÁ: ( NOP –áí Ýß³Ý³Ï»Ýù ¹³ï³ñÏ ·áñÍáÕáõÃÛáõÝÁ, x-ÇÝ y-Ç ³ñÅ»ùÁ í»ñ³·ñ»Éáõ Ñ³Ù³ñ Ïû·ï³·áñÍ»Ýù x←y Ýß³Ý³ÏáõÙÁ:)

1. x⇐ AVAIL – ÏáõÛïÇó í»ñóÝ»É Ýáñ Ñ³Ý·áõÛó ¨ Ýñ³ óáõóÇãÁ ·ñ³Ýó»É x-áõÙ: ²Ûë ·áñÍáÕáõÃÛáõÝÁ Çñ³Ï³Ý³óíáõÙ ¿ Ñ»ï¨Û³É Ï»ñå.

**»Ã»** (AVAIL = NULL)

**³å³** OVERFLOW IN HEAP

**³ÛÉ³å»ë** { x ← AVAIL, AVAIL ← LINK (AVAIL)}

1. x ⇒ AVAIL – Ñ³Ý·áõÛóÁ í»ñ³¹³ñÓÝ»É ÏáõÛï: ²Ûë ·áñÍáÕáõÃÛáõÝÁ Çñ³Ï³Ý³óíáõÙ ¿ Ñ»ï¨Û³É Ï»ñå.

{ LINK(x) ← AVAIL, AVAIL ← x }

**¶Í³ÛÇÝ óáõó³ÏÇ Çñ³Ï³Ý³óáõÙÁ ½³Ý·í³ÍÇ ÙÇçáóáí**

¸Çóáõù x1. . . . xn –Á óáõó³ÏÁ Ý»ñÏ³Û³óÝáÕ ½³Ý·í³ÍÝ ¿, ÇëÏ m-Á óáõó³ÏÇ »ñÏ³ñáõÃÛáõÝÁ (0<=m<=n):

ÜÏ³ñ³·ñ»Ýù óáõó³ÏÇ Ñ»ï Ï³ï³ñíáÕ Ñ»ï¨Û³É ·áñÍáÕáõÃÛáõÝÝ»ñÁ:

1. InsertAt(p, item) - item ï³ññÇ ³í»É³óáõÙ óáõó³ÏÇ p-ñ¹ ¹ÇñùáõÙ: ²Ûë ¹»åùáõÙ ½³Ý·í³ÍÇ µáÉáñ ï³ññ»ñÁ ëÏë³Í p -ñ¹ ¹ÇñùÇó ï»Õ³÷áËíáõÙ »Ý Ù»Ï ¹Çñùáí ¹»åÇ ³ç:

1 m n

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

p ÜÏ. 1.3

**»Ã»** (m=n)

**³å³** OVERFLOW

**³ÛÉ³å»ë**

**»Ã»** (p<1 Ï³Ù p>m+1)

**³å³** NOP

**³ÛÉ³å»ë** { x[i+1] ← x[i], (i=m,m-1,…,p), x[p] ← item, m←m+1}

1. Delete(p) - ï³ññÇ Ñ»é³óáõÙ p -ñ¹ ¹ÇñùÇó: ²Ûë ¹»åùáõÙ ½³Ý·í³ÍÇ µáÉáñ ï³ññ»ñÁ (p +1)-ñ¹ ¹ÇñùÇó ëÏë³Í ï»Õ³÷áËíáõÙ »Ý Ù»Ï ¹Çñùáí ¹»åÇ Ó³Ë:

**»Ã»** (m=0)

**³å³** UNDERFLOW

**³ÛÉ³å»ë**

**»Ã»** (p<1 Ï³Ù p>m)

**³å³** NOP

**³ÛÉ³å»ë** { x[i] ← x[i+1], (i=p,p+1,…,m-1), m←m-1}

Æñ³Ï³Ý³óáõÙ Ï³å³Ïóí³Í óáõó³ÏÝ»ñÇ ÙÇçáóáí

¶Í³ÛÇÝ óáõó³ÏÁ Ï³ñ»ÉÇ ¿ Çñ³Ï³Ý³óÝ»É ·Í³ÛÇÝ ÙÇ³Ï³å, óÇÏÉÇÏ ¨ »ñÏÏ³å óáõó³ÏÝ»ñÇ ÙÇçáóáí: êïáñ¨ µ»ñí³Í »Ý ³Û¹åÇëÇ óáõó³ÏÝ»ñÇ ï»ëù»ñÁ ¨ ÝÏ³ñ³·ñí³Í »Ý Ýñ³Ýó Ñ»ï Ï³ï³ñíáÕ áñáß ·áñÍáÕáõÃÛáõÝÝ»ñ:

**¶Í³ÛÇÝ óáõó³ÏÇ Çñ³Ï³Ý³óáõÙÁ ÙÇ³Ï³å ·Í³ÛÇÝ óáõó³ÏÇ ÙÇçáóáí**

Î³å³Ïóí³Í óáõó³ÏÝ»ñÇ å³ñ½³·áõÛÝ ¹»åùÁ ÙÇ³Ï³å ·Í³ÛÇÝ óáõó³ÏÝ ¿: ØÇ³Ï³å ·Í³ÛÇÝ óáõó³ÏáõÙ Ûáõñ³ù³ÝãÛáõñ Ñ³Ý·áõÛó Ï³å ¿ å³Ñå³ÝáõÙ óáõó³ÏáõÙ Çñ»Ý Ñ³çáñ¹áÕ Ñ³Ý·áõÛóÇ Ñ»ï: ì»ñçÇÝ Ñ³Ý·áõÛóÇ Ï³åÇ ¹³ßïáõÙ ·ñíáõÙ ¿ NULL:

ØÇ³Ï³å ·Í³ÛÇÝ óáõó³ÏÁ Ï³ñ»ÉÇ ¿ å³ïÏ»ñ»É Ñ»ï¨Û³É Ó¨áí.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  | **NULL** |

F P

ÜÏ. 1.4

²Ûëï»Õ F–Á óáõóÇã ¿ óáõó³ÏÇ ³é³çÇÝ Ñ³Ý·áõÛóÇ íñ³, ÇëÏ P–Ý óáõóÇã ¿ áñ¨¿ Ñ³Ý·áõÛóÇ íñ³:

ÜÏ³ñ³·ñ»Ýù ÙÇ³Ï³å óáõó³ÏÇ Ñ»ï Ï³ï³ñíáÕ Ñ»ï¨Û³É ·áñÍáÕáõÃÛáõÝÝ»ñÁ:

1. InsertAfter(P, item) - óáõó³ÏáõÙ P(P≠Λ) óáõóÇãáí Ñ³Ý·áõÛóÇó Ñ»ïá item ï³ññÇ ³í»É³óáõÙ:

x⇐AVAIL, INFO(x) ←item, LINK(x) ←LINK(P), LINK(P) ←x

2. InsertAt(n,item) - item ï³ññÇ ³í»É³óáõÙ óáõó³ÏÇ n-ñ¹ ¹ÇñùáõÙ: ¸Çñù»ñÁ Ñ³Ù³ñ³Ï³Éí³Í »Ý ëÏë³Í ½ñáÛÇó:

**»Ã»** (n=0)

**³å³** {x ⇐ AVAIL,INFO(x) ← item, LINK(x) ← F, F ← x}

**³ÛÉ³å»ë** {P←F,

P←LINK (P) ( i=0,1,…, n-1),

InsertAfter (P, item)}

3. DeleteAfter(P) - P(P≠Λ) óáõóÇãáí Ñ³Ý·áõÛóÇÝ Ñ³çáñ¹áÕ Ñ³Ý·áõÛóÇ Ñ»é³óáõÙ:

**»Ã»** ( LINK(P) = 0)

**³å³** NOP

**³ÛÉ³å»ë** {x ← LINK(P), LINK(P) ← LINK(x), x ⇒ AVAIL }

**¶Í³ÛÇÝ óáõó³ÏÇ Çñ³Ï³Ý³óáõÙÁ óÇÏÉÇÏ óáõó³ÏÇ ÙÇçáóáí**

òÇÏÉÇÏ óáõó³Ï Ï³Ýí³Ý»Ýù ³ÛÝ ÙÇ³Ï³å ·Í³ÛÇÝ óáõó³ÏÁ, áñÇ í»ñçÇÝ Ñ³Ý·áõÛóÇ Ï³åÇ ¹³ßïáõÙ ·ñí³Í ¿ ³é³çÇÝ Ñ³Ý·áõÛóÇ Ñ³ëó»Ý:

PTR

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |

ÜÏ. 1.5

²Ûëï»Õ PTR-Á óáõóÇã ¿ óÇÏÉÇÏ óáõó³ÏÇ áñ¨¿ Ñ³Ý·áõÛóÇ íñ³:

ÜÏ³ñ³·ñ»Ýù óÇÏÉÇÏ óáõó³ÏÇ Ñ»ï Ï³ï³ñíáÕ Ñ»ï¨Û³É ·áñÍáÕáõÃÛáõÝÝ»ñÁ:

1. Insertleft (item) - óÇÏÉÇÏ óáõó³ÏáõÙ PTR óáõóÇãáí Ñ³Ý·áõÛóÇó Ñ»ïá item ï³ññÇ ³í»É³óáõÙ:

x⇐AVAIL, INFO (x) ← item,

**»Ã»** (PTR = NULL)

**³å³** {LINK (x) ←x, PTR←x}

**³ÛÉ³å»ë** {LINK(x) ← LINK(PTR), LINK(PTR) ← x}

1. InsertRight(item) - óÇÏÉÇÏ óáõó³ÏáõÙ PTR óáõóÇãáí Ñ³Ý·áõÛóÇó Ñ»ïá item ï³ññÇ ³í»É³óáõÙ ¨ PTR óáõóãÇ ï»Õ³÷áËáõÙ ³í»É³óíáÕ Ñ³Ý·áõÛóÇ íñ³:

InsertLeft(item), PTR ← x

1. DeleteAfter() - óÇÏÉÇÏ óáõó³ÏÇó PTR óáõóÇãáí Ñ³Ý·áõÛóÇÝ Ñ³çáñ¹áÕ ï³ññÇ Ñ»é³óáõÙ:

**»Ã»** (PTR = NULL)

**³å³** UNDERFLOW

**³ÛÉ³å»ë** { x←LINK (PTR), LINK(PTR) ←LINK(x),

**»Ã»** (PTR = x) **³å³** PTR← NULL,

x ⇒ AVAIL }

1. ClearList() - óáõó³ÏÇ µáÉáñ ï³ññ»ñÇ Ñ»é³óáõÙ:

**»Ã»** (PTR = NULL)

**³å³** NOP

**³ÛÉ³å»ë** { P← AVAIL, AVAIL← LINK(PTR), LINK(PTR) ←P, PTR← NULL }

ø³ÝÇ áñ óÇÏÉÇÏ óáõó³ÏáõÙ í»ñçÇÝ Ñ³Ý·áõÛóÇ Ï³åÇ ¹³ßïÁ ¹³ï³ñÏ ã¿, ³é³ç³ÝáõÙ ¿ óáõó³ÏÇ í»ñçÁ áñáß»Éáõ ËÝ¹Çñ: ²Ûë ¹»åùáõÙ å»ïù ¿ ß³ñÅí»É óáõó³ÏÇ íñ³Ûáí` ³ÝóÝ»Éáí ÙÇ Ñ³Ý·áõÛóÇó ÙÛáõëÇÝ, ³ÛÝù³Ý Å³Ù³Ý³Ï, ù³ÝÇ ¹»é ã»Ýù Ñ³ë»É óáõó³ÏÇ ëÏ½µÝ³Ï³Ý ¹ÇñùÇÝ (³ÛÝ Ñ³Ý·áõÛóÇÝ, áñÇó ëÏëí»É ¿ñ ï³ññ»ñÇ ¹Çï³ñÏáõÙÁ):

²Ûë ËÝ¹ÇñÁ Ï³ñ»ÉÇ ¿ ÉáõÍ»É Ý³¨ óÇÏÉÇÏ óáõó³ÏáõÙ ³í»É³óÝ»Éáí Ñ³ïáõÏ Ýßí³Í Ñ³Ý·áõÛó, áñÁ Ï³Ý·³éÇ Ñ³Ù³ñ Ñ³ñÙ³ñ ¹Çñù ÏÑ³Ý¹Çë³Ý³: òáõó³ÏÇ ³Û¹ Ñ³Ý·áõÛóÁ ÏáãíáõÙ ¿ í»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛó: ²Û¹åÇëÇ óáõó³ÏÝ»ñÇ ³é³í»ÉáõÃÛáõÝÝ»ñÇó Ù»ÏÝ ³ÛÝ ¿, áñ Ýñ³Ýù »ñµ»ù ¹³ï³ñÏ ã»Ý ÉÇÝáõÙ: ì»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛóáí óÇÏÉÇÏ óáõó³ÏÁ å³ïÏ»ñí³Í ¿ ÝÏ. 1.6 –áõÙ:

í»ñÝ³·ñ³ÛÇÝ PTR

Ñ³Ý·áõÛó

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

ÜÏ. 1.6

¸³ï³ñÏ óáõó³ÏÁ ÏáõÝ»Ý³ Ñ»ï¨Û³É ï»ëùÁ`

í»ñÝ³·ñ³ÛÇÝ PTR

Ñ³Ý·áõÛó

|  |  |
| --- | --- |
|  |  |

**¶Í³ÛÇÝ óáõó³ÏÇ Çñ³Ï³Ý³óáõÙÁ »ñÏÏ³å óáõó³ÏÇ ÙÇçáóáí**

Î³å³Ïóí³Í ·Í³ÛÇÝ óáõó³ÏÝ»ñÇ ï³ñ³ï»ë³ÏÝ»ñÇó ¿ »ñÏÏ³å ·Í³ÛÇÝ óáõó³ÏÁ: ºñÏÏ³å ·Í³ÛÇÝ óáõó³ÏáõÙ Ûáõñ³ù³ÝãÛáõñ Ñ³Ý·áõÛó Ï³å ¿ å³Ñå³ÝáõÙ Ý³Ëáñ¹ ¨ Ñ³çáñ¹ Ñ³Ý·áõÛóÝ»ñÇ Ñ»ï` Ñ³Ù³å³ï³ëË³Ý³µ³ñ LLINK ¨ RLINK óáõóÇãÝ»ñÇ ÙÇçáóáí:

ì»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛóáí »ñÏÏ³å ·Í³ÛÇÝ óáõó³ÏÁ å³ïÏ»ñí³Í ¿ ÝÏ. 1.7-áõÙ: HEAD-Á í»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛóÇ óáõóÇãÝ ¿:

í»ñÝ³·ñ³ÛÇÝ HEAD

Ñ³Ý·áõÛó LLINK RLINK

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

ÜÏ. 1.7

¸³ï³ñÏ óáõó³ÏÝ ³Ûë ¹»åùáõÙ ÏáõÝ»Ý³ Ñ»ï¨Û³É ï»ëùÁ`

í»ñÝ³·ñ³ÛÇÝ HEAD

Ñ³Ý·áõÛó

|  |  |  |
| --- | --- | --- |
|  |  |  |

ÜÏ³ï»Ýù, áñ »ñÏÏ³å óáõó³ÏÇ µáÉáñ Ñ³Ý·áõÛóÝ»ñÇ Ñ³Ù³ñ ï»ÕÇ áõÝÇ Ñ»ï¨Û³É å³ÛÙ³ÝÁ.

RLINK(LLINK(P))= LLINK(RLINK(P))=P, áñï»Õ P–Ý Ñ³Ý·áõÛóÇ óáõóÇã ¿:

ÜÏ³ñ³·ñ»Ýù í»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛóáí »ñÏÏ³å óáõó³ÏÇ Ñ»ï Ï³ï³ñíáÕ Ñ»ï¨Û³É ·áñÍáÕáõÃÛáõÝÝ»ñÁ:

1. IsEmpty() - óáõó³ÏÇ ¹³ï³ñÏ ÉÇÝ»Éáõ ëïáõ·áõÙ:

**»Ã»** (LLINK(HEAD)=HEAD)

**³å³** TRUE

**³ÛÉ³å»ë** FALSE

1. InsertRight(P, item) - óáõó³ÏáõÙ P (P≠Λ) óáõóÇãáí Ñ³Ý·áõÛóÇó Ñ»ïá item ï³ññÇ ³í»É³óáõÙ :

x⇐AVAIL, INFO(x) ←item, RLINK(x) ←RLINK(P),

LLINK(x) ←P, y ←RLINK(P), RLINK(P) ←x, LLINK(y) ←x

1. Delete(P) - P (P≠Λ) óáõóÇãáí Ñ³Ý·áõÛóÇ Ñ»é³óáõÙ:

**»Ã»** ( P = HEAD)

**³å³** UNDERFLOW

**³ÛÉ³å»ë** {

RLINK(LLINK(P))← RLINK(P),

LLINK(RLINK(P))← LLINK(P),

P ⇒ AVAIL

}

2.3 ¶Í³ÛÇÝ óáõó³ÏÇ Çñ³Ï³Ý³óáõÙÁ C++É»½íáí

Ìñ³·ñ³íáñÙ³Ý É»½áõÝ»ñáõÙ ïíÛ³ÉÝ»ñÇ ³µëïñ³Ïï ïÇå»ñÁ Çñ³Ï³Ý³óÝ»Éáõ Ñ³Ù³ñ ë³ÑÙ³ÝíáõÙ »Ý ïíÛ³ÉÝ»ñÇ Ýáñ ïÇå»ñ, áñáÝù Çñ³Ï³Ý³óíáõÙ »Ý ¹³ë»ñÇ ÙÇçáóáí: C++É»½íáõÙ LIst ¹³ëÁ Ý»ñÏ³Û³óÝ»Ýù ß³µÉáÝÇ ï»ëùáí, áñï»Õ áñå»ë ß³µÉáÝÇ å³ñ³Ù»ïñ Ñ³Ù³ñáõÙ »Ýù óáõó³ÏÇ ï³ññ»ñÇ ïÇåÁ: ÜÏ. 1.8-áõÙ Ý»ñÏ³Û³óí³Í ¿ LIst ¹³ëÇ ß³µÉáÝÇ UML-¹Ç³·ñ³ÙÁ: ºÝÃ³¹ñíáõÙ ¿, áñ ïíÛ³ÉÝ»ñÇ Ñ³çáñ¹³Ï³ÝáõÃÛáõÝÝ Çñ³Ï³Ý³óí³Í ¿ ÙÇ³Ï³å ·Í³ÛÇÝ óáõó³ÏÇ ÙÇçáóáí:



ÜÏ.1.8

LIst ¹³ëÇ Ñ³Ûï³ñ³ñáõÙÁ Ï³ï³ñ»Ýù List.h ý³ÛÉáõÙ:

//òáõó³ÏÇ Ñ³Ý·áõÛóÇ Node ïÇåÁ Ñ³Ûï³ñ³ñáõÙ

template <class T>

struct Node

{ T info;

Node<T>\* link;

};

//LIst ¹³ëÇ Ñ³Ûï³ñ³ñáõÙ

template<class T>

class List

{

private:

Node< T >\*first;

int size;

public:

List (); //ÉéáõÃÛ³Ùµ ÏáÝëïñáõÏïáñ

List (const List&); // å³ï×»ÝÇ ÏáÝëïñáõÏïáñ

~ List (); //¹»ëïñáõÏïáñ

//¶áñÍáÕáõÃÛáõÝÝ»ñ

bool IsEmpty() const;

int GetLenght() const;

Node< T >\* GetFirst() const;

void InsertAfter(Node< T >\*p, T item);

void InsertAt(int n, T item);

void DeleteAfter (Node< T >\*p);

void DeleteAt (int n);

T & GetData(Node< T >\*p);

};

êïáñ¨ µ»ñí³Í ¿ LIst ¹³ëÇ Çñ³Ï³Ý³óÙ³Ý List.cpp ý³ÛÉÁ :

#include “List.h”

//ÈéáõÃÛ³Ùµ ÏáÝëïñáõÏïáñ

template <class T>

List <T>::List ():size(0), first(NULL)

{

}

//ä³ï×»ÝÇ ÏáÝëïñáõÏïáñ

template <class T>

LIst <T>::List(const List& aList)

{

//óáõó³ÏÇ ³é³çÇÝ Ñ³Ý·áõÛóÇ å³ï×»ÝáõÙ

Node <T>\* p=new Node <T>, p1= aList.first;

GetData(p)=aList.GetData(p1);

first=p; first→link=NULL;

//óáõó³ÏÇ ÙÝ³ó³Í Ñ³Ý·áõÛóÝ»ñÇ å³ï×»ÝáõÙ

‎for(int i=0; i< aList.GetLenght() -1; i++)

{

p1=p1→link;

InsertAfter (p, aList.GetData(p1));

p=p→link;

}

size= aList.size;

}

//¸»ëïñáõÏïáñ

template <class T>

List <T>::~List ()

{

if (first)

{while (first → link)

DeleteAfter(first);

delete first;

first=0;

}

size=0;

}

//IsEmpty() –ëïáõ·áõÙ ¿ óáõó³ÏÇ ¹³ï³ñÏ ÉÇÝ»ÉÁ

template <class T>

bool List <T>:: IsEmpty() const

{

return size==0;

}

//GetLenght() - í»ñ³¹³ñÓÝáõÙ ¿ óáõó³ÏÇ ï³ññ»ñÇ ù³Ý³ÏÁ

template <class T>

int List <T>:: GetLenght() const

{

return size;

}

//GetFirst() - í»ñ³¹³ñÓÝáõÙ ¿ óáõóÇã óáõó³ÏÇ ³é³çÇÝ ï³ññÇ íñ³

template <class T>

Node<T>\* List <T>:: GetFirst () const

{

return first;

}

//InsertAfter(p,item) –óáõó³ÏáõÙ p óáõóÇãáí Ñ³Ý·áõÛóÇó Ñ»ïá ³í»É³óÝáõÙ ¿ item ï³ññÁ

template <class T>

void List <T>::InsertAfter (Node< T >\*p, T item)

{

Node <T>\* q=new Node<T>;

q→info=item;

q→link=p→link;

p→link=q;

}

//InsertAt(n,item**)** – óáõó³ÏÇ n–ñ¹ ¹ÇñùáõÙ ³í»É³óÝáõÙ ¿ item ï³ññÁ

template <class T>

void List <T>::InsertAt (int n, T item)

{

if (n<0 || i> GetLenght()) throw “position is out of range”;

if (n==0)

{Node <T>\* q=new Node<T>;

q→info=item;

q→link=first;

first=q;

return;

}

Node <T>\*p=first;

‎for(int i=0;i<n-1;i++)

p=p→link;

InsertAfter (p,item);

}

//DeleteAfter(p) - óáõó³ÏÇó Ñ»é³óÝáõÙ ¿ p óáõóÇãáí Ñ³Ý·áõÛóÇÝ Ñ³çáñ¹áÕ ï³ññÁ

template <class T>

void List <T>:: DeleteAfter(Node< T >\*p)

{

if (IsEmpty())

{ throw “List is empty”;

}

if (p->link==NULL)

{throw ”No element to delete”;

}

Node <T>\*q=p->link;

p->link=q→link;

delete q;

return;

}

//DeleteAt(n) - Ñ»é³óÝáõÙ ¿ óáõó³ÏÇ n-ñ¹ ¹ÇñùáõÙ ·ïÝíáÕ ï³ññÁ

template <class T>

void List <T>:: DeleteAt(int n)

{

if (n<0 || n>=GetLenght()) throw “position is out of range”;

if (n==0 && GetLenght()!=0)

{Node <T>\*q=first;

first= first→link;

delete q;

return;

}

Node <T>\*p=first;

‎for(int i=0;i<n-1;i++)

p=p→link;

DeleteAfter (p,item);

}

//GetData(p) – í»ñ³¹³ñÓÝáõÙ ¿ óáõó³ÏÇ p óáõóÇãáí Ñ³Ý·áõÛóÇ ï³ññÁ

template <class T>

T& List <T>::GetData(Node< T >\*p)

{

return p→info;

}

´»ñ»Ýù LIst ¹³ëÝ û·ï³·áñÍáÕ Íñ³·ñÇ ûñÇÝ³Ï:

# include <iostream.h>

# include “List.h

int main()

{ try

{

const int n=10;

List <double> L1; // ëï»ÕÍíáõÙ ¿ ¹³ï³ñÏ óáõó³Ï

//L1 óáõó³ÏÁ ³ñÅ»ù³íáñ»É ëï»ÕÝ³ß³ñÇó Ý»ñÙáõÍí³Í double ïÇåÇ ï³ññ»ñáí

double a[n]; int I;

for(i=0; i<n-1; i++)

{ cin>> a[i]; L1.InsertAt(i, a[i]);}

Node< double >\* p;

List <double> L2=L1; //L2 óáõó³ÏÁ ³ñÅ»ù³íáñíáõÙ ¿ L1 óáõó³ÏÇ ï³ññ»ñáí

for( i=0, p= L2.GetFirst();i<L2.GetLenght();i++,p=p->link)

cout<< L2.GetData(p)<<’ ‘; //³ñï³ÍíáõÙ »Ý L2 óáõó³ÏÇ ï³ññ»ñÁ

} //try µÉáÏÇ ³í³ñï

//µ³ó³éÇÏ Çñ³íÇ×³ÏÇ Ùß³ÏáõÙ

catch (char\* a)

{cout<<a<<endl;}

return 0;

}

2.5 ¶Í³ÛÇÝ óáõó³ÏÇ ÏÇñ³éáõÃÛáõÝÝ»ñ

´³½Ù³Ý¹³ÙÝ»ñÇ ·áõÙ³ñáõÙ

²ÙµáÕç ·áñÍ³ÏÇóÝ»ñáí µ³½Ù³Ý¹³ÙÝ»ñÇ Ñ³Ù³ñ ¹Çï³ñÏ»Ýù ¹ñ³Ýó ·áõÙ³ñÙ³Ý ·áñÍáÕáõÃÛ³Ý Çñ³Ï³Ý³óáõÙÁ: ´³½Ù³Ý¹³ÙÝ»ñÁ Ý»ñÏ³Û³óÝ»Ýù í»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛóáí Ï³å³Ïóí³Í óÇÏÉÇÏ óáõó³ÏÝ»ñÇ ï»ëùáí, áñÇ Ñ³Ý·áõÛóÝ»ñÁ µ³½Ù³Ý¹³ÙÇ áã ½ñáÛ³Ï³Ý ³Ý¹³ÙÝ»ñÝ »Ý ¨ áõÝ»Ý Ñ»ï¨Û³É Ï³éáõóí³ÍùÁ`

|  |  |  |
| --- | --- | --- |
| COEF | PWR | LINK |

²Ûëï»Õ COEF-Á ÙÇ³Ý¹³ÙÇ ·áñÍ³ÏÇóÝ ¿, ÇëÏ PWR –Á` ³ëïÇ×³ÝÁ: ºÝÃ³¹ñíáõÙ ¿, áñ µ³½Ù³Ý¹³ÙÇ ³Ý¹³ÙÝ»ñÝ ÁÝÃ³ÝáõÙ »Ý ³ëïÇ×³ÝÝ»ñÇ Ýí³½Ù³Ý Ï³ñ·áí:

úñÇÝ³Ï, 5x4-4x3+x-7 µ³½Ù³Ý¹³ÙÁ Ý»ñÏ³Û³óíáõÙ ¿ Ñ»ï¨Û³É Ï»ñå.

PTR

|  |  |  |
| --- | --- | --- |
| 5 | 4 |  |

|  |  |  |
| --- | --- | --- |
| -4 | 3 |  |

|  |  |  |
| --- | --- | --- |
| 1 | 1 |  |

|  |  |  |
| --- | --- | --- |
| -7 | 0 |  |

|  |  |  |
| --- | --- | --- |
| 0 | -1 |  |

²Ûëï»Õ, PTR-áí Ýßí³Í í»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛóÇ PWR ¹³ßïáõÙ ·ñí³Í –1 ³ñÅ»ùÝ ³å³ÑáíáõÙ ¿ ³É·áñÇÃÙÇ µÝ³Ï³ÝáÝ ³í³ñïÁ:

**²É·áñÇÃÙ (´³½Ù³Ý¹³ÙÝ»ñÇ ·áõÙ³ñáõÙ):** ²Ûë ³É·áñÇÃÙÁ (P) µ³½Ù³Ý¹³ÙÁ ·áõÙ³ñáõÙ ¿ (Q) µ³½Ù³Ý¹³ÙÇÝ, »ÝÃ³¹ñ»Éáí, áñ P-Ý ¨ Q-Ý óáõóÇãÝ»ñ »Ý, áñáÝù ÑÕíáõÙ »Ý í»ñÁ Ýßí³Í ï»ëùáí µ³½Ù³Ý¹³ÙÝ»ñÇÝ: P óáõó³ÏÁ ãÇ ÷áËíáõÙ, ÇëÏ Q-áõÙ ëï³óíáõÙ ¿ ·áõÙ³ñÁ: ²É·áñÇÃÙÇ ³ßË³ï³ÝùÇ í»ñçáõÙ óáõóã³ÛÇÝ ÷á÷áË³Ï³ÝÝ»ñÝ ÁÝ¹áõÝáõÙ »Ý Çñ»Ýó ëÏ½µÝ³Ï³Ý ³ñÅ»ùÝ»ñÁ: ú·ï³·áñÍíáõÙ »Ý Ý³¨ Q1 ¨ Q2 Éñ³óáõóÇã óáõóÇãÝ»ñ:

**Øáõïù** - (P) ¨ (Q) µ³½Ù³Ý¹³ÙÝ»ñ

**ºÉù** - (P) ¨ (Q) µ³½Ù³Ý¹³ÙÝ»ñÇ ·áõÙ³ñ

**ø³ÛÉ 1**. [êÏ½µÝ³Ï³Ý ï»Õ³¹ñáõÃÛáõÝÝ»ñ] P ← LINK(P), Q1 ← Q, Q ← LINK(Q):

²ÛÅÙ P-Ý ¨ Q-Ý ÝßáõÙ »Ý(ÑÕíáõÙ »Ý) µ³½Ù³Ý¹³ÙÝ»ñÇ ³í³· ³Ý¹³ÙÝ»ñÇÝ: ²Ûë ³É·áñÇÃÙáõÙ Ñ³Ù³ñÛ³ ³Ù»Ýáõñ»ù Q1 ÷á÷áË³Ï³ÝÁ “Ù»Ï ù³ÛÉáí Ñ»ï ¿ ÙÝáõÙ” Q-Çó, ³ÛëÇÝùÝ` Q=LINK(Q1):

**ø³ÛÉ 2**. [Ð³Ù»Ù³ï»É PWR(P)-Ý ¨ PWR(Q)-Ý ]

ºÃ» (PWR(P) < PWR(Q)), ³å³ {Q1← Q, Q ← LINK(Q), ³ÝóÝ»É **ø³ÛÉ 2**}:

ºÃ» (PWR(P) = PWR(Q)), ³å³ ³ÝóÝ»É **ø³ÛÉ 3:**

ºÃ» (PWR(P) > PWR(Q)), ³å³ ³ÝóÝ»É **ø³ÛÉ 5:**

**ø³ÛÉ 3**. [¶áõÙ³ñ»É ·áñÍ³ÏÇóÝ»ñÁ] ¶ïÝí»É »Ý Ñ³í³ë³ñ ³ëïÇ×³ÝÝ»ñáí »ñÏáõ ³Ý¹³ÙÝ»ñ:

ºÃ» (PWR(P)<0), ³å³ ³É·áñÇÃÙÝ ³í³ñï»É, ³ÛÉ³å»ë COEF(Q)←COEF(Q)+COEF(P):

ºÃ» (COEF(Q)=0), ³å³ ³ÝóÝ»É **ø³ÛÉ 4,**³ÛÉ³å»ë{Q1←Q,P←LINK(P), Q ← LINK(Q), ³ÝóÝ»É **ø³ÛÉ 2}**:

**ø³ÛÉ 4**. [Ð»é³óÝ»É 0-³Ï³Ý ³Ý¹³ÙÁ]

Q2← Q, LINK(Q1) ← Q ← LINK(Q), AVAIL⇐ Q2, P← LINK(P), ³ÝóÝ»É **ø³ÛÉ 2**:

**ø³ÛÉ 5**. [²í»É³óÝ»É Ýáñ ³Ý¹³Ù](P) µ³½Ù³Ý¹³ÙÁ å³ñáõÝ³ÏáõÙ ¿ ³ÛÝåÇëÇ óáõóÇãáí ³Ý¹³Ù, áñÁ µ³ó³Ï³ÛáõÙ ¿ (Q)-áõÙ: ²Û¹ ³Ý¹³ÙÝ ³í»É³óíáõÙ ¿ (Q)-ÇÝ:

Q2⇐ AVAIL, COEF(Q2) ← COEF(P), PWR(Q2) ← PWR(P), LINK(Q2) ← Q, LINK(Q1) ← Q2, Q1← Q2, P← LINK(P), ³ÝóÝ»É **ø³ÛÉ 2**:

***ÊÝ¹ÇñÝ»ñ***

1. îñí³Í ¿ ³ÙµáÕç Ãí»ñÇó Ï³½Ùí³Í Ñ³çáñ¹³Ï³ÝáõÃÛáõÝ: Æñ³Ï³Ý³óÝ»É ³Û¹ Ñ³çáñ¹³Ï³ÝáõÃÛáõÝÇó ïñí³Í ï³ññÇ Ñ»é³óÙ³Ý ¨ ³í»É³óÙ³Ý ·áñÍáÕáõÃÛáõÝÝ»ñÁ:
2. îñí³Í ¿ ³ÙµáÕç Ãí»ñÇó Ï³½Ùí³Í Ï³ñ·³íáñí³Í Ñ³çáñ¹³Ï³ÝáõÃÛáõÝ: Æñ³Ï³Ý³óÝ»É ³Û¹ Ñ³çáñ¹³Ï³ÝáõÃÛáõÝÇó ïñí³Í ï³ññÇ µÇÝ³ñ ÷ÝïñÙ³Ý ³É·áñÇÃÙÁ:
3. Î³å³Ïóí³Í óáõó³ÏÇ ï»ëùáí Ý»ñÏ³Û³óí³Í µ³½Ù³Ý¹³ÙÇ Ñ³Ù³ñ Çñ³Ï³Ý³óÝ»É µ³½Ù³Ý¹³ÙÇ ³ñÅ»ùÇ Ñ³ßíáõÙÁ ïñí³Í Ï»ïáõÙ:
4. Î³å³Ïóí³Í óáõó³ÏÇ ï»ëùáí Ý»ñÏ³Û³óí³Í µ³½Ù³Ý¹³ÙÇ Ñ³Ù³ñ Çñ³Ï³Ý³óÝ»É µ³½Ù³Ý¹³ÙÝ»ñÇ Ñ³ÝáõÙÁ:
5. Î³å³Ïóí³Í óáõó³ÏÇ ï»ëùáí Ý»ñÏ³Û³óí³Í µ³½Ù³Ý¹³ÙÝ»ñÇ Ñ³Ù³ñ Çñ³Ï³Ý³óÝ»É Ýñ³Ýó µ³½Ù³å³ïÏÙ³Ý ³É·áñÇÃÙÁ:
6. ´³½ÙáõÃÛáõÝÁ Ý»ñÏ³Û³óíáõÙ ¿ ÙÇ³Ï³å ·Í³ÛÇÝ óáõó³ÏÇ ï»ëùáí: Úáõñ³ù³ÝãÛáõñ Ñ³Ý·áõÛóáõÙ å³ÑíáõÙ ¿ µ³½ÙáõÃÛ³Ý Ù»Ï ï³ññ: ´³½ÙáõÃÛ³Ý ï³ññ»ñÁ óáõó³ÏáõÙ ¹³ë³íáñí³Í »Ý ³×Ù³Ý Ï³ñ·áí:

³. îñí³Í »ñÏáõ µ³½ÙáõÃÛáõÝÝ»ñÇ Ñ³Ù³ñ Ï³éáõó»É Ýñ³Ýó ÙÇ³íáñáõÙÁ:

µ. îñí³Í »ñÏáõ µ³½ÙáõÃÛáõÝÝ»ñÇ Ñ³Ù³ñ Ï³éáõó»É Ýñ³Ýó Ñ³ïáõÙÁ:

1. ´³½Ù³Ý¹³ÙÁ Ý»ñÏ³Û³óíáõÙ ¿ í»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛóáí ÙÇ³Ï³å ·Í³ÛÇÝ óáõó³ÏÇ ï»ëùáí: Úáõñ³ù³ÝãÛáõñ Ñ³Ý·áõÛóáõÙ ÝßíáõÙ ¿ Ñ»ñÃ³Ï³Ý ÙÇ³Ý¹³ÙÇ ·áñÍ³ÏÇóÁ, ³ëïÇ×³ÝÁ ¨ Ï³åÁ Ñ³çáñ¹ Ñ³Ý·áõÛóÇ Ñ»ï: ²ëïÇ×³ÝÝ»ñÁ ¹³ë³íáñí³Í »Ý ÙÇ³Ý¹³ÙÝ»ñÇ ³ëïÇ×³ÝÝ»ñÇ Ýí³½Ù³Ý Ï³ñ·áí: Æñ³Ï³Ý³óÝ»É »ñÏáõ µ³½Ù³Ý¹³ÙÝ»ñÇ ·áõÙ³ñÙ³Ý ¨ µ³½Ù³Ý¹³ÙÇ ³ñï³ÍÙ³Ý ·áñÍáÕáõÃÛáõÝÝ»ñÁ:

3. ä³ÑáõÝ³Ï

3.1 ä³ÑáõÝ³Ï ïíÛ³ÉÝ»ñÇ ³µëïñ³Ïï ïÇåÁ

ä³ÑáõÝ³ÏÁ ³é³í»É Ñ³×³Ë û·ï³·áñÍíáÕ ¨ Ï³ñ¨áñ ïíÛ³ÉÝ»ñÇ Ï³éáõóí³Íù ¿: ä³ÑáõÝ³ÏÁ û·ï³·áñÍíáõÙ ¿, ûñÇÝ³Ï, ÏáÙåÇÉÛ³ïáñÝ»ñáõÙ Íñ³·ñ»ñÇ ß³ñ³ÑÛáõëáõÃÛáõÝÁ ëïáõ·»ÉÇë, ³ñï³Ñ³ÛïáõÃÛáõÝÝ»ñÇ ³ñÅ»ùÝ»ñÁ Ñ³ßí»ÉÇë, ýáõÝÏóÇ³Ý»ñÇ Ï³Ýã»ñÁ Çñ³Ï³Ý³óÝ»ÉÇë ¨ ³ÛÉÝ: ä³ÑáõÝ³ÏÁ å³ñáõÝ³ÏáõÙ ¿ ï³ññ»ñ, ¨ áõÝÇ ·³·³ÃÁ ÝßáÕ ¹Çñù (Top): ä³ÑáõÝ³ÏÇ Ñ»ï Ï³ï³ñíáÕ ÑÇÙÝ³Ï³Ý ·áñÍáÕáõÃÛáõÝÝ»ñÝ »Ý ï³ññÇ ³í»É³óáõÙÁ (Push) ¨ Ñ»é³óáõÙÁ (Pop), áñáÝù Çñ³Ï³Ý³óíáõÙ »Ý å³ÑáõÝ³ÏÇ ·³·³ÃÇó: ä³ÑáõÝ³ÏáõÙ í»ñçÇÝ ³í»É³óñ³Í ï³ññÁ Ñ»é³óíáõÙ ¿ ³é³çÇÝÁ: ²Û¹ å³ï×³éáí å³ÑáõÝ³ÏÇ Ñ³Ù³ñ ³ëáõÙ »Ý, áñ Ý³ áõÝÇ LIFO Ï³ñ· (last-in , first-out` í»ñçÇÝÁ »Ï³í ³é³çÇÝÁ ¹áõñë »Ï³í): ºñµ å³ÑáõÝ³ÏáõÙ ³ÛÉ¨ë ÑÝ³ñ³íáñ ã¿ ï³ññ ³í»É³óÝ»É, ³ëáõÙ »Ý, áñ å³ÑáõÝ³ÏÁ ÉÇùÝ ¿, ÇëÏ »ñµ ÑÝ³ñ³íáñ ã¿ ï³ññ Ñ»é³óÝ»É å³ÑáõÝ³ÏÁ ¹³ï³ñÏ ¿ Ñ³Ù³ñíáõÙ:

ADT Stack

*îíÛ³ÉÝ»ñ*

î³ññ»ñÇ óáõó³Ï` å³ÑáõÝ³ÏÇ ·³·³ÃÁ ÝßáÕ Top ¹Çñùáí:

*¶áñÍáÕáõÃÛáõÝÝ»ñ*

ÎáÝëïñáõÏïáñ

Ý³Ë³å³ÛÙ³Ý - ãÏ³

åñáó»ë - å³ÑáõÝ³ÏÇ ëÏ½µÝ³ñÅ»ù³íáñáõÙ

Empty

Ùáõïù - ãÏ³

Ý³Ë³å³ÛÙ³Ý - ãÏ³

åñáó»ë - ëïáõ·áõÙ, ¹³ï³ñÏ ¿ ³ñ¹Ûáù å³ÑáõÝ³ÏÁ

»Éù - true, »Ã» å³ÑáõÝ³ÏÁ ¹³ï³ñÏ ¿, false` Ñ³Ï³é³Ï ¹»åùáõÙ

í»ñçÝ³å³ÛÙ³Ý - ãÏ³

Push

Ùáõïù – å³ÑáõÝ³ÏÇÝ ³í»É³óíáÕ ï³ññÁ

Ý³Ë³å³ÛÙ³Ý – ãÏ³

åñáó»ë – ï³ññÇ ³í»É³óáõÙ å³ÑáõÝ³ÏÇÝ

»Éù – ãÏ³

í»ñçÝ³å³ÛÙ³Ý – ³í»É³óí³Í ï³ññáí å³ÑáõÝ³Ï

µ³ó³éÇÏ Çñ³íÇ×³Ï – Ýáñ ï³ññÇ ³í»É³óáõÙ Éóí³Í å³ÑáõÝ³ÏÇÝ

Pop

Ùáõïù - ãÏ³

Ý³Ë³å³ÛÙ³Ý – å³ÑáõÝ³ÏÁ ¹³ï³ñÏ ã¿

åñáó»ë – ï³ññÇ Ñ»é³óáõÙ å³ÑáõÝ³ÏÇó

»Éù – å³ÑáõÝ³ÏÇ ·³·³ÃÇ ï³ññÁ

í»ñçÝ³å³ÛÙ³Ý – Ñ»é³óí³Í ï³ññáí å³ÑáõÝ³Ï

µ³ó³éÇÏ Çñ³íÇ×³Ï – ï³ññÇ Ñ»é³óáõÙ ¹³ï³ñÏ å³ÑáõÝ³ÏÇó

Peek

Ùáõïù - ãÏ³

Ý³Ë³å³ÛÙ³Ý – å³ÑáõÝ³ÏÁ ¹³ï³ñÏ ã¿

åñáó»ë - å³ÑáõÝ³ÏÇ ·³·³ÃÇ ï³ññÇ ÁÝïñáõÙ

»Éù - å³ÑáõÝ³ÏÇ ·³·³ÃÇ ï³ññÁ

í»ñçÝ³å³ÛÙ³Ý – å³ÑáõÝ³ÏÝ ³Ý÷á÷áË ¿

µ³ó³éÇÏ Çñ³íÇ×³Ï – ï³ññÇ ÁÝïñáõÙ ¹³ï³ñÏ å³ÑáõÝ³ÏÇó

ClearStack

Ùáõïù - ãÏ³

Ý³Ë³å³ÛÙ³Ý - ãÏ³

åñáó»ë - å³ÑáõÝ³ÏÇ µáÉáñ ï³ññ»ñÇ Ñ»é³óáõÙ

»Éù – ãÏ³

í»ñçÝ³å³ÛÙ³Ý – ¹³ï³ñÏ å³ÑáõÝ³Ï

# í»ñç ADT Stack

3.2 ä³ÑáõÝ³ÏÇ Çñ³Ï³Ý³óáõÙ

ä³ÑáõÝ³ÏÁ, ÇÝãå»ë ¨ ·Í³ÛÇÝ óáõó³ÏÁ, Ï³ñ»ÉÇ ¿ Çñ³Ï³Ý³óÝ»É »ñÏáõ »Õ³Ý³Ïáí` ½³Ý·í³ÍÇ ¨ Ï³å³Ïóí³Í óáõó³ÏÇ ÙÇçáóáí:

**ä³ÑáõÝ³ÏÇ Çñ³Ï³Ý³óáõÙÁ ½³Ý·í³ÍÇ ÙÇçáóáí**

ºÝÃ³¹ñ»Ýù áõÝ»Ýù x1 . . . . xm ½³Ý·í³ÍÁ ¨ Top ÷á÷áË³Ï³ÝÁ, áñáí ïñíáõÙ ¿ å³ÑáõÝ³ÏÇ ·³·³ÃÇ ÇÝ¹»ùëÁ` 0≤Top≤m:

ÜÏ³ñ³·ñ»Ýù å³ÑáõÝ³ÏÇ Ñ»ï Ï³ï³ñíáÕ ·áñÍáÕáõÃÛáõÝÝ»ñÁ:

1. Empty() - ëïáõ·»É å³ÑáõÝ³ÏÇ ¹³ï³ñÏ ÉÇÝ»ÉÁ:

**»Ã»** (Top = 0 )

**³å³** TRUE

**³ÛÉ³å»ë** FALSE

1. Push(y) - å³ÑáõÝ³ÏáõÙ ³í»É³óÝ»É»É Ýáñ ï³ññ:

**»Ã»** (Top = m)

**³å³** OVERFLOW

**³ÛÉ³å»ë** { Top ← Top + 1, x[Top] ← y}

1. Pop() - å³ÑáõÝ³ÏÇó Ñ»é³óÝ»É ï³ññ:

**»Ã»** (Top = 0)

**³å³** UNDERFLOW

**³ÛÉ³å»ë** Top ← Top – 1

1. GetTop(y) - ëï³Ý³É å³ÑáõÝ³ÏÇ ·³·³ÃÇ ï³ññÁ:

**»Ã»** (Top = 0)

**³å³** UNDERFLOW

**³ÛÉ³å»ë** y ← x[Top]

1. ClearStack() - å³ÑáõÝ³ÏÁ ¹³ï³ñÏ»É:

Top ← 0

**ä³ÑáõÝ³ÏÇ Çñ³Ï³Ý³óáõÙÁ Ï³å³Ïóí³Í óáõó³ÏÇ ÙÇçáóáí**

|  |  |
| --- | --- |
| INFO |  |

|  |  |
| --- | --- |
| INFO |  |

|  |  |
| --- | --- |
| INFO | NULL |

. . . .

Top

ÜÏ. 3.1

²Ûëï»Õ Top – Á å³ÑáõÝ³ÏÇ ·³·³ÃÇ óáõóÇãÝ ¿: ºÃ» å³ÑáõÝ³ÏÁ ¹³ï³ñÏ ¿, ³å³ óáõóÇãÝ áõÝÇ NULL ³ñÅ»ùÁ:

ÜÏ³ñ³·ñ»Ýù å³ÑáõÝ³ÏÇ Ñ»ï Ï³ï³ñíáÕ ·áñÍáÕáõÃÛáõÝÝ»ñÁ:

1. Empty() - ëïáõ·»É å³ÑáõÝ³ÏÇ ¹³ï³ñÏ ÉÇÝ»ÉÁ:

**»Ã»** (Top = NULL)

**³å³** TRUE

**³ÛÉ³å»ë** FALSE

1. Push(y) - å³ÑáõÝ³ÏáõÙ ³í»É³óÝ»É Ýáñ ï³ññ:

x ⇐ AVAIL, LINK(x) ← Top, INFO(x) ← y , Top ← x

1. Pop() - å³ÑáõÝ³ÏÇó Ñ»é³óÝ»É ï³ññ:

**»Ã»** (Top = NULL)

**³å³** UNDERFLOW

**³ÛÉ³å»ë** {P ← Top, Top ← LINK(Top), p ⇒ AVAIL}

1. GetTop(y) – ëï³Ý³É å³ÑáõÝ³ÏÇ ·³·³ÃÇ ï³ññÁ:

**»Ã»** (Top = NULL)

**³å³** UNDERFLOW

**³ÛÉ³å»ë** y ← INFO(Top)

1. ClearStack() - å³ÑáõÝ³ÏÁ ¹³ï³ñÏ»É:

Top ← NULL

3.3 ä³ÑáõÝ³ÏÇ Çñ³Ï³Ý³óáõÙÁ C++ É»½íáí

êïáñ¨ Ý»ñÏ³Û³óí³Í »Ý å³ÑáõÝ³ÏÇ Çñ³Ï³Ý³óÙ³Ý ï³ñµ»ñ »Õ³Ý³ÏÝ»ñ C++ É»½íáí:

1. ä³ÑáõÝ³ÏÇ Ñ³çáñ¹³Ï³Ý Çñ³Ï³Ý³óáõÙ:

ä³ÑáõÝ³ÏÝ Çñ³Ï³Ý³óí³Í ¿ Stack ¹³ëÇ ÙÇçáóáí: ÜÏ. 3.2-áõÙ Ý»ñÏ³Û³óí³Í ¿ Stack ¹³ëÇ UML-¹Ç³·ñ³ÙÁ:



ÜÏ. 3.2

²ÛÅÙ Ñ³çáñ¹³Ï³Ý å³ÑáõÝ³ÏÝ Çñ³Ï³Ý³óÝ»Ýù LIFO ¹³ëÇ ß³µÉáÝÇ ÙÇçáóáí, áñï»Õ áñå»ë ß³µÉáÝÇ å³ñ³Ù»ïñ Ñ³Ù³ñáõÙ »Ýù å³ÑáõÝ³ÏÇ ï³ññ»ñÇ ïÇåÁ ¨ å³ÑáõÝ³ÏÇ ï³ññ»ñÁ å³Ñ»Éáõ Ñ³Ù³ñ Ý³Ë³ï»ëí³Í ½³Ý·í³ÍÇ ã³÷Á: Èóí³Í å³ÑáõÝ³ÏÇÝ Ýáñ ï³ññ ³í»É³óÝ»Éáõ ¨ ¹³ï³ñÏ å³ÑáõÝ³ÏÇó ï³ññ Ñ»é³óÝ»Éáõ ÷áñÓ»ñ ³Ý»Éáõ ¹»åùáõÙ ·»Ý»ñ³óíáõÙ ¿ µ³ó³éÇÏ Çñ³íÇ×³Ï:

LIFO ¹³ëÇ ß³µÉáÝÇ Ñ³Ûï³ñ³ñáõÙÁ Ï³ï³ñ»Ýù LifoTemplate.h ý³ÛÉáõÙ:

template<class T, int maxsize>

class LIFO

{

private:

T st\_mas[maxsize];

int top; // å³ÑáõÝ³ÏÇ ·³·³Ã

public:

LIFO():top(-1){}

~LIFO(){}

bool IsEmpty(){return top==-1;}

bool IsFull(){return top==maxsize-1;}

void push(T item);

void pop();

T& getTop();

void print();

};

êïáñ¨ µ»ñí³Í ¿ LIFO ¹³ëÇ ß³µÉáÝÇ Çñ³Ï³Ý³óáõÙÁ :

template <class T, int maxsize>

void LIFO <T, maxsize>::push(T item)

{

if(IsFull()) throw "LIFO is full";

else

st\_mas[++top]=item;

}

template <class T, int maxsize>

void LIFO <T, maxsize>::pop()

{

if(IsEmpty()) throw "LIFO is empty";

else

top--;

}

template <class T, int maxsize>

T LIFO <T, maxsize>:: getTop()

{

if(IsEmpty()) throw "LIFO is empty";

return st\_mas[top];

}

template <class T, int maxsize>

void LIFO <T, maxsize>:: print()

{

for(int i=0;i<top+1;i++)

cout<<st\_mas[i];

cout<<endl;

}

¸Çï³ñÏ»Ýù å³ÑáõÝ³ÏÇ û·ï³·áñÍÙ³Ý Ñ»ï¨Û³É ûñÇÝ³ÏÁ:

int main()

{

try

{

LIFO <int,5> x;

cout<<"Is empty? "<<x.IsEmpty()<<endl;

x.push(2); x.push(3); x.print();

cout<<"Is empty? "<<x.IsEmpty()<<endl;

x.pop(); x.print();

cout<<"top "<<x.getTop()<<endl;

LIFO <char \*,50> y;

cout<<y.IsEmpty()<<endl;

y.push(“First element”); y.push(“First element”);

cout<<y.IsEmpty()<<endl;

cout<<"top "<<y.getTop()<<endl;

}

catch(LIFO\_Exception & a)

{a.show();}

return 0;

}

2. ä³ÑáõÝ³ÏÇ Ï³å³Ïóí³Í Çñ³Ï³Ý³óáõÙ:

Æñ³Ï³Ý³óÝ»Ýù å³ÑáõÝ³ÏÁ Ï³å³Ïóí³Í óáõó³ÏÇ ÙÇçáóáí: Î³éáõó»Ýù LIFO ¹³ëÇ ß³µÉáÝÁ, áñï»Õ áñå»ë ß³µÉáÝÇ å³ñ³Ù»ïñ Ñ³Ù³ñáõÙ »Ýù å³ÑáõÝ³ÏÇ ï³ññ»ñÇ ïÇåÁ: òáõó³ÏÇ Ñ³Ý·áõÛóÁ ÝÏ³ñ³·ñíáõÙ ¿ Node ïÇåáí: Èóí³Í å³ÑáõÝ³ÏÇÝ Ýáñ ï³ññ ³í»É³óÝ»Éáõ ¨ ¹³ï³ñÏ å³ÑáõÝ³ÏÇó ï³ññ Ñ»é³óÝ»Éáõ ÷áñÓ»ñ ³Ý»Éáõ ¹»åùáõÙ ·»Ý»ñ³óíáõÙ ¿ µ³ó³éÇÏ Çñ³íÇ×³Ï, áñÁ Ý»ñÏ³Û³óíáõÙ ¿ LIFO\_Exception ûµÛ»ÏïÇ ÙÇçáóáí:

#include<iostream>

#include<string>

#include<exception>

using namespace std;

//LIFO\_Exception ¹³ë

class LIFO\_Exception: public exception

{

private:

char \* name;

public:

LIFO\_Exception(){}

LIFO\_Exception(char \*s)

{

name=new char[strlen(s)+1];

strcpy(name,s);

}

~LIFO\_Exception(){delete []name;}

void show(){cout<<name<<endl;}

};

//LIFO ¹³ëÇ ß³µÉáÝ

template<class T>

struct Node

{

T info;

Node <T>\* link;

};

template<class T>

class LIFO

{

private:

Node<T>\* top; //top-Á óáõóÇã ¿ å³ÑáõÝ³ÏÇ ·³·³ÃÇ ï³ññÇ íñ³

public:

LIFO():top(0){}

~LIFO(){}

bool IsEmpty(){return top==0;}

//bool IsFull(){return top==maxsize-1;}

void push(T item);

void pop();

T getTop();

void print();

};

template <class T>

void LIFO <T>::push(T item)

{

Node<T>\* ptr=new Node<T>; // ëï³Ý³É Ýáñ Ñ³Ý·áõÛó ÏáõÛïÇó

If (!ptr) throw LIFO\_Exception("Overload in heap");

ptr->info=item;

ptr->link=top;

top=ptr;

}

template <class T>

T LIFO <T>:: getTop()

{

if(IsEmpty()) throw LIFO\_Exception("LIFO is empty");

return top->info;

}

template <class T>

void LIFO <T>:: pop()

{

if(IsEmpty()) throw LIFO\_Exception("LIFO is empty");

Node<T>\* ptr=top;

top=top->link;

delete ptr;

return;

}

template <class T>

void LIFO <T>::print()

{

Node<T>\* ptr=top;

while(ptr!=0)

{cout<<ptr->info<<' ';

ptr=ptr->link;

}

cout<<endl;

}

ÆÝãå»ë ³ñ¹»Ý Ýßí»É ¿, å³ÑáõÝ³ÏÁ ÏÇñ³éíáõÙ ¿ Ãí³µ³Ý³Ï³Ý ³ñï³Ñ³ÛïáõÃÛáõÝÝ»ñÇ Ý»ñÏ³Û³óÙ³Ý ï³ñµ»ñ Ó¨»ñÝ Çñ³Ï³Ý³óÝ»ÉÇë, ÙÇ Ó¨Çó ÙÛáõëÇÝ ³ÝóÝ»ÉÇë ¨ ³ñï³Ñ³ÛïáõÃÛáõÝÝ»ñÇ ³ñÅ»ùÝ»ñÁ Ñ³ßí»ÉÇë:

¸Çï³ñÏ»Ýù Ãí³µ³Ý³Ï³Ý ³ñï³Ñ³ÛïáõÃÛáõÝÝ»ñÇ Ý»ñÏ³Û³óÙ³Ý ÙÇç³Í³Ýó³ÛÇÝ **(infix)**, Ý³Ë³Í³Ýó³ÛÇÝ **(prefix)**, í»ñç³Í³Ýó³ÛÇÝ **(postfix)** Ó¨»ñÁ:

**ØÇç³Í³Ýó³ÛÇÝ Ó¨**

²ñï³Ñ³ÛïáõÃÛ³Ý Ý»ñÏ³Û³óÙ³Ý ÙÇç³Í³Ýó³ÛÇÝ Ó¨áõÙ ·áñÍáÕáõÃÛ³Ý Ýß³ÝÁ ¹ñíáõÙ ¿ ûå»ñ³Ý¹Ý»ñÇ ÙÇç¨, ûñÇÝ³Ï` 2+3\*5:

²ñï³Ñ³ÛïáõÃÛáõÝÝ»ñÇ Ý»ñÏ³Û³óÙ³Ý ÙÇç³Í³Ýó³ÛÇÝ Ó¨áõÙ Ï³ñ¨áñíáõÙ »Ý ·áñÍáÕáõÃÛáõÝÝ»ñÇ Ý³Ë³å³ïíáõÃÛáõÝÝ»ñÁ (priority), áñáÝù áñáßáõÙ »Ý ·áñÍáÕáõÃÛáõÝÝ»ñÇ Ï³ï³ñÙ³Ý Ï³ñ·Á: àñáß ·áñÍáÕáõÃÛáõÝÝ»ñ áõÝ»Ý Ñ³Ù³ñÅ»ù Ý³Ë³å³ïíáõÃÛáõÝÝ»ñ: ²Û¹ ¹»åùáõÙ ¹ñ³Ýù Ï³ï³ñíáõÙ »Ý Ó³ËÇó ¹»åÇ ³ç Ï³Ù ³çÇó ¹»åÇ Ó³Ë: êïáñ¨ Ãí³ñÏíáõÙ »Ý ·áñÍáÕáõÃÛáõÝÝ»ñÁ Áëï Ý³Ë³å³ïíáõÃÛáõÝÝ»ñÇ Ýí³½Ù³Ý`

* ³ëïÇ×³Ý µ³ñÓñ³óÝ»Éáõ ·áñÍáÕáõÃÛáõÝ (↑)
* µ³½Ù³å³ïÏÙ³Ý(\*) ¨ µ³Å³ÝÙ³Ý(/) ·áñÍáÕáõÃÛáõÝÝ»ñ
* ·áõÙ³ñÙ³Ý(+) ¨ Ñ³ÝÙ³Ý(-) ·áñÍáÕáõÃÛáõÝÝ»ñ

¶áñÍáÕáõÃÛáõÝÝ»ñÇ Ï³ï³ñÙ³Ý Ï³ñ·Á ÷áËáõÙ »Ý ÷³Ï³·Í»ñÇ ÙÇçáóáí:

**Ü³Ë³Í³Ýó³ÛÇÝ Ó¨**

²Ûë ¹»åùáõÙ ·áñÍáÕáõÃÛ³Ý Ýß³ÝÁ ¹ñíáõÙ ¿ ûå»ñ³Ý¹Ý»ñÇó ³é³ç:

¸Çóáõù AωB ÙÇç³Í³Ýó³ÛÇÝ ïÇåÇ ³ñï³Ñ³ÛïáõÃÛáõÝ ¿, áñï»Õ ω-Ý ·áñÍáÕáõÃÛ³Ý Ýß³Ý ¿, ÇëÏ A-Ý ¨ B-Ý ³Û¹ ·áñÍáÕáõÃÛ³Ý ûå»ñ³Ý¹Ý»ñÝ »Ý:

prefix (AωB) = ωprefix(A)prefix(B)

prefix (A) = A, áñï»Õ A-Ý ÃÇí ¿ Ï³Ù ÷á÷áË³Ï³Ý:

úñÇÝ³Ï.

prefix(a+b)= +ab

prefix(a+b\*c)= +a\*bc

prefix(a\*(b+c)↑2-5) = -prefix(a\*(b+c)↑2)prefix(5)= - \* prefix(a)prefix((b+c)↑2)5 = - \* a↑prefix(b+c)prefix(2)5 = - \* a↑+bc25

**ì»ñç³Í³Ýó³ÛÇÝ Ó¨**

²ñï³Ñ³ÛïáõÃÛ³Ý Ý»ñÏ³Û³óÙ³Ý í»ñç³Í³Ýó³ÛÇÝ Ó¨áõÙ ûå»ñ³Ý¹Ý»ñÁ Ý³Ëáñ¹áõÙ »Ý ·áñÍáÕáõÃÛáõÝÝ»ñÇÝ: ²Û¹ ·ñ»É³Ó¨Á ³Ýí³ÝáõÙ »Ý Ý³¨ É»Ñ³Ï³Ý Ý»ñÏ³Û³óáõÙ` RPN (Reverse Polish Notation):

postfix(AωB) = postfix(A)postfix(B)ω,

postfix(A) = A, áñï»Õ A-Ý ÃÇí ¿ Ï³Ù ÷á÷áË³Ï³Ý:

úñÇÝ³Ï.

postfix(a+b)= ab+

postfix(a+b\*c)= abc\*+

postfix(a\*(b+c)↑2-5) = postfix(a(b+c)↑2)postfix(5) - = postfix(a) postfix((b+c)↑2)\*5- = aposfix(b+c)postfix(2)↑\*5 - = abc+2↑\*5-

Ü³Ë³Í³Ýó³ÛÇÝ ¨ í»ñç³Í³Ýó³ÛÇÝ Ó¨»ñáõÙ ÷³Ï³·Í»ñ ã»Ý û·ï³·áñÍíáõÙ:

**²ñï³Ñ³ÛïáõÃÛ³Ý Ó¨³÷áËáõÙÁ ÙÇç³Í³Ýó³ÛÇÝ Ó¨Çó í»ñç³Í³Ýó³ÛÇÝ Ó¨ÇÝ**

ÜÏ³ñ³·ñ»Ýù ³ñï³Ñ³ÛïáõÃÛ³Ý ÙÇç³Í³Ýó³ÛÇÝ Ó¨Çó í»ñç³Í³Ýó³ÛÇÝÇÝ Ó¨³÷áËÙ³Ý ³É·áñÇÃÙÁ:

ØÇç³Í³Ýó³ÛÇÝ ³ñï³Ñ³ÛïáõÃÛáõÝÁ ïñíáõÙ ¿ S ïáÕáí, áñÁ Ï³ñáÕ ¿ Ï³ñ¹³óí»É ëï»ÕÝ³ß³ñÇó Ï³Ù ï»ùëï³ÛÇÝ ý³ÛÉÇó: ²É·áñÇÃÙÇ ³ßË³ï³ÝùÇ ³ñ¹ÛáõÝùáõÙ í»ñç³Í³Ýó³ÛÇÝ Ó¨Á ëï³óíáõÙ ¿ U ïáÕáõÙ, áñÁ Ï³ñáÕ ¿ ³ñï³Íí»É ¿Ïñ³ÝÇÝ Ï³Ù ï»ùëï³ÛÇÝ ý³ÛÉáõÙ: priority(ω) ‎ýáõÝÏóÇ³Ý í»ñ³¹³ñÓÝáõÙ ¿ ω ·áñÍáÕáõÃÛ³Ý Ý³Ë³å³ïíáõÃÛáõÝÁ: ú·ï³·áñÍíáõÙ ¿ ·áñÍáÕáõÃÛáõÝÝ»ñÇ opstk å³ÑáõÝ³ÏÁ:

**Øáõïù** - ÙÇç³Í³Ýó³ÛÇÝ ³ñï³Ñ³ÛïáõÃÛáõÝÁ S ïáÕáõÙ

**ºÉù** - í»ñç³Í³Ýó³ÛÇÝ ³ñï³Ñ³ÛïáõÃÛáõÝÁ U ïáÕáõÙ

**ø³ÛÉ 1**. [ëÏ½µÝ³ñÅ»ù³íáñáõÙ] ¹³ï³ñÏ»É opstk å³ÑáõÝ³ÏÁ, ëï»ÕÍ»É ¹³ï³ñÏ U ïáÕ

**ø³ÛÉ 2.** [Ñ»ñÃ³Ï³Ý ëÇÙíáÉÇ Ý»ñÙáõÍáõÙ] S-Çó Ï³ñ¹³óíáÕ Ñ»ñÃ³Ï³Ý ëÇÙíáÉÁ

·ñíáõÙ ¿ c ÷á÷áË³Ï³ÝáõÙ S → c;

**ø³ÛÉ 3.** [ÙáõïùÇ í»ñç]

**»Ã»** (S–Ç µáÉáñ ëÇÙíáÉÝ»ñÁ Ï³ñ¹³óí»É »Ý)

**³å³** ³ÝóÝ»É ø**³ÛÉ 5**

**ø³ÛÉ 4**. [c-Ç ëïáõ·áõÙ]

**»Ã»** (c-Ý ûå»ñ³Ý¹ ¿)

**³å³** {c → U (c-Ý ·ñ»É U-áõÙ), ³ÝóÝ»É **ø³ÛÉ 2** }

**»Ã»** (c-Ý Ñ³í³ë³ñ ¿ ‘ ( ‘ )

**³å³** {c ⇒ opstk , ³ÝóÝ»É **ø³ÛÉ 2**}

**»Ã»** (c-Ý ·áñÍáÕáõÃÛ³Ý Ýß³Ý ¿)

**³å³**

{ **ù³ÝÇ ¹»é** (opstk-Á ¹³ï³ñÏ ã¿ ¨ ·³·³ÃÇ ï³ññÁ Ñ³í³ë³ñ ã¿ ‘ (‘ )

{ x ⇐ opstk;

**»Ã»** (priority(x) ≥ priority(c))

**³å³** x→U

**³ÛÉ³å»ë** {x ⇒ opstk, »Éù óÇÏÉÇó }

}

c ⇒ opstk;

}

**»Ã»** (c-Ý Ñ³í³ë³ñ ¿ ‘ ) ‘)

**³å³** { x ⇐ opstk

**ù³ÝÇ ¹»é** (x ≠ ‘( ‘)

{ x → U, x ⇐ opstk }

}

³ÝóÝ»É **ø³ÛÉ 2**

**ø³ÛÉ 5**. [¹³ï³ñÏ»É opstk -Á]

**ù³ÝÇ ¹»é** (opstk != Λ)

{ x⇐ opstk, x → U }

**ì»ñç³Í³Ýó³ÛÇÝ Ó¨áí ïñí³Í ³ñï³Ñ³ÛïáõÃÛ³Ý ³ñÅ»ùÇ Ñ³ßíáõÙÁ**

¸Çï³ñÏ»Ýù í»ñç³Í³Ýó³ÛÇÝ Ó¨áí ïñí³Í ³ñï³Ñ³ÛïáõÃÛ³Ý ³ñÅ»ùÇ Ñ³ßíÙ³Ý ³É·áñÇÃÙÁ: ì»ñç³Í³Ýó³ÛÇÝ ïáÕáõÙ Ñ³Ý¹Çå³Í ·áñÍáÕáõÃÛáõÝÁ Ï³ï³ñíáõÙ ¿ Çñ»Ý Ý³Ëáñ¹áÕ »ñÏáõ ûå»ñ³Ý¹Ý»ñÇ ÝÏ³ïÙ³Ùµ: úå»ñ³Ý¹Ý»ñÇó Ù»ÏÁ Ï³ñáÕ ¿ ÉÇÝ»É Ý³Ëáñ¹ ·áñÍáÕáõÃÛ³Ý ³ñ¹ÛáõÝù:

ì»ñç³Í³Ýó³ÛÇÝ ³ñï³Ñ³ÛïáõÃÛáõÝÁ ïñíáõÙ ¿ U ïáÕáí, áñÁ Ï³ñáÕ ¿ Ï³ñ¹³óí»É ëï»ÕÝ³ß³ñÇó Ï³Ù ï»ùëï³ÛÇÝ ý³ÛÉÇó: ú·ï³·áñÍíáõÙ ¿ ûå»ñ³Ý¹Ý»ñÇ stk å³ÑáõÝ³ÏÁ: U-Çó Ï³ñ¹³óíáÕ Ñ»ñÃ³Ï³Ý ûå»ñ³Ý¹Á ·ñíáõÙ ¿ stk å³ÑáõÝ³ÏáõÙ: ºÃ» U-Çó Ï³ñ¹³óíáõÙ ¿ ·áñÍáÕáõÃÛ³Ý Ýß³Ý, ³å³ ·áñÍáÕáõÃÛáõÝÁ Ï³ï³ñíáõÙ ¿ å³ÑáõÝ³ÏÇ ·³·³ÃÇ »ñÏáõ ûå»ñ³Ý¹Ý»ñÇ ÝÏ³ïÙ³Ùµ ¨ ëï³óí³Í ³ñ¹ÛáõÝùÁ ÑÇßíáõÙ ¿ å³ÑáõÝ³ÏáõÙ:

²É·áñÇÃÙÇ ³ßË³ï³ÝùÇ ³ñ¹ÛáõÝùáõÙ å³ÑáõÝ³ÏáõÙ ëï³óíáõÙ ¿ ïñí³Í ³ñï³Ñ³ÛïáõÃÛ³Ý ³ñÅ»ùÁ:

**Øáõïù** - í»ñç³Í³Ýó³ÛÇÝ ³ñï³Ñ³ÛïáõÃÛáõÝÁ U ïáÕáõÙ

**ºÉù** - ³ñï³Ñ³ÛïáõÃÛ³Ý ³ñÅ»ùÁ x ÷á÷áË³Ï³ÝáõÙ

**ø³ÛÉ 1**. [ëÏ½µÝ³ñÅ»ù³íáñáõÙ]

¹³ï³ñÏ»É stk å³ÑáõÝ³ÏÁ

**ø³ÛÉ 2**. [Ñ»ñÃ³Ï³Ý ëÇÙíáÉÇ Ý»ñÙáõÍáõÙ U ïáÕÇó]

U → c

**ø³ÛÉ 3**. [ÙáõïùÇ í»ñç]

**»Ã»** (U-Ç µáÉáñ ëÇÙíáÉÝ»ñÁ Ï³ñ¹³óí»É »Ý)

**³å³** ³ÝóÝ»É **ø³ÛÉ 5**

**ø³ÛÉ 4**. [ëÇÙíáÉÇ Ùß³ÏáõÙ]

**»Ã»** (c-Ý ûå»ñ³Ý¹ ¿)

**³å³** c ⇒ stk

**»Ã»** (c-Ý ω ·áñÍáÕáõÃÛ³Ý Ýß³Ý ¿)

**³å³**

{ x ⇐ stk, y ⇐ stk, z=y ω x, z ⇒ stk }

³ÝóÝ»É **ø³ÛÉ 2**

**ø³ÛÉ 5**. [³í³ñï]

x ⇐ stk

***ÊÝ¹ÇñÝ»ñ***

1. \* Æñ³Ï³Ý³óÝ»É å³ÑáõÝ³ÏÁ ¹ÇÝ³ÙÇÏ ½³Ý·í³ÍÇ ÙÇçáóáí:
2. Æñ³Ï³Ý³óÝ»É å³ÑáõÝ³Ï, ¹ÇÝ³ÙÇÏ ½³Ý·í³ÍÇ û·ÝáõÃÛ³Ùµ: ä³ÑáõÝ³ÏÇ ·»ñÑ³·»óÙ³Ý ¹»åùáõÙ ÏñÏÝ³ÏÇ ³í»É³óÝ»É ½³Ý·í³ÍÇ ã³÷Á:
3. Æñ³Ï³Ý³óÝ»É »ñÏáõ å³ÑáõÝ³ÏÝ»ñÇ DoubleStack ¹³ëÁ, áñÝ û·ï³·áñÍáõÙ ¿ ½³Ý·í³Í: ØÇ å³ÑáõÝ³ÏÁ ï»Õ³¹ñíáõÙ ¿ ½³Ý·í³ÍÇ ëÏ½µÇó, ÇëÏ ÙÛáõëÁ` í»ñçÇó: ä³ÑáõÝ³ÏÝ»ñÇ ·³·³ÃÝ»ñÇ ÇÝ¹»ùëÝ»ñÁ ïñíáõÙ »Ý Top1 ¨ Top2 ³ÙµáÕç ïÇåÇ ÷á÷áË³Ï³ÝÝ»ñáí: ²Ûë »ñÏáõ å³ÑáõÝ³ÏÝ»ñÇ Ñ³Ù³ñ Çñ³Ï³Ý³óÝ»É å³ÑáõÝ³ÏÝ»ñÇÝ µÝáñáß µáÉáñ ·áñÍáÕáõÃÛáõÝÝ»ñÁ:
4. \* îñí³Í »Ý ³ÙµáÕç Ãí»ñÇ S1 , ëÇÙíáÉÝ»ñÇ S2 ¨ Çñ³Ï³Ý Ãí»ñÇ S3 å³ÑáõÝ³ÏÝ»ñÁ: Ø»ÏáõÙ»ç ïå»É ³Û¹ å³ÑáõÝ³Ï»ñÇ ï³ññ»ñÁ ÙÇÝã¨ µáÉáñ å³ÑáõÝ³Ï»ñÇ ¹³ï³ñÏí»ÉÁ:

4. Ð»ñÃ

4.1 Ð»ñÃ ïíÛ³ÉÝ»ñÇ ³µëïñ³Ïï ïÇåÁ

Ð»ñÃÁ (Queue) ·Í³ÛÇÝ óáõó³Ï ¿, áñï»Õ ïíÛ³ÉÝ»ñÇ Ñ³ë³Ý»ÉÇáõÃÛáõÝÁ Çñ³Ï³Ý³óíáõÙ ¿ óáõó³ÏÇ »ñÏáõ »½ñ»ñÇó: î³ññÝ ³í»É³óíáõÙ ¿ Ñ»ñÃÇ í»ñçáõÙ (Ñ»ñÃÇ í»ñç` rear) ¨ Ñ»é³óíáõÙ ¿ Ñ»ñÃÇ ëÏ½µÇó (Ñ»ñÃÇ ëÏÇ½µ` front): Ð»ñÃÇó ï³ññ»ñÁ Ñ»é³óíáõÙ »Ý ³ÛÝ Ï³ñ·áí ÇÝãå»ë å³Ñí³Í »Ý, Ñ»ï¨³µ³ñ Ñ»ñÃÁ ³å³ÑáíáõÙ ¿ FIFO Ï³ñ· (first-in / first-out` ³é³çÇÝÁ »Ï³í ³é³çÇÝÁ ¹áõñë »Ï³í):

ADT Queue

îíÛ³ÉÝ»ñ

ï³ññ»ñÇ óáõó³Ï

front – Ñ»ñÃáõÙ ³é³çÇÝ ï³ññÇ ¹ÇñùÁ

rear – Ñ»ñÃáõÙ í»ñçÇÝ ï³ññÇÝ Ñ³çáñ¹áÕ ¹ÇñùÁ

count – Ñ»ñÃÇ ï³ññ»ñÇ ù³Ý³ÏÁ

¶áñÍáÕáõÃÛáõÝÝ»ñ

ÎáÝëïñáõÏïáñ

Ùáõïù - ãÏ³

Ý³Ë³å³ÛÙ³Ý – ãÏ³

åñáó»ë - Ñ»ñÃÇ ëÏ½µÝ³ñÅ»ù³íáñáõÙ

í»ñçÝ³å³ÛÙ³Ý – ¹³ï³ñÏ Ñ»ñÃ

Length

Ùáõïù – ãÏ³

Ý³Ë³å³ÛÙ³Ý – ãÏ³

åñáó»ë – Ñ»ñÃÇ ï³ññ»ñÇ ù³Ý³ÏÇ áñáßáõÙ

»Éù – Ñ»ñÃÇ ï³ññ»ñÇ ù³Ý³ÏÁ

í»ñçÝ³å³ÛÙ³Ý – ãÏ³

Empty

Ùáõïù – ãÏ³

Ý³Ë³å³ÛÙ³Ý – ãÏ³

åñáó»ë – Ñ»ñÃÇ ¹³ï³ñÏ ÉÇÝ»Éáõ ëïáõ·áõÙ

»Éù – true, »Ã» Ñ»ñÃÁ ¹³ï³ñÏ ¿, false` Ñ³Ï³é³Ï ¹»åùáõÙ

í»ñçÝ³å³ÛÙ³Ý – ãÏ³

Insert

Ùáõïù - Ñ»ñÃÇÝ ³í»É³óíáÕ ï³ññÁ

Ý³Ë³å³ÛÙ³Ý – ãÏ³

åñáó»ë – ï³ññÇ ³í»É³óáõÙ Ñ»ñÃÇÝ

»Éù – ãÏ³

í»ñçÝ³å³ÛÙ³Ý – ³í»É³óí³Í ï³ññáí Ñ»ñÃ

µ³ó³éÇÏ Çñ³íÇ×³Ï – ï³ññÇ ³í»É³óáõÙ Éóí³Í Ñ»ñÃÇÝ

Delete

Ùáõïù – ãÏ³

Ý³Ë³å³ÛÙ³Ý – Ñ»ñÃÁ ¹³ï³ñÏ ã¿

åñáó»ë – ï³ññÇ Ñ»é³óáõÙ Ñ»ñÃÇó

»Éù – ãÏ³

í»ñçÝ³å³ÛÙ³Ý – Ñ»é³óí³Í ï³ññáí Ñ»ñÃ

µ³ó³éÇÏ Çñ³íÇ×³Ï – ï³ññÇ Ñ»é³óáõÙ ¹³ï³ñÏ Ñ»ñÃÇó

Front

Ùáõïù – ãÏ³

Ý³Ë³å³ÛÙ³Ý - Ñ»ñÃÁ ¹³ï³ñÏ ã¿

åñáó»ë – Ñ»ñÃÇ ëÏ½µÇ ï³ññÇ ÁÝïñáõÙ

»Éù – Ñ»ñÃÇ ëÏ½µÇ ï³ññÁ

í»ñçÝ³å³ÛÙ³Ý – ãÏ³

µ³ó³éÇÏ Çñ³íÇ×³Ï – ï³ññÇ ÁÝïñáõÙ ¹³ï³ñÏ Ñ»ñÃÇó

ClearQueue

Ùáõïù – ãÏ³

Ý³Ë³å³ÛÙ³Ý - ãÏ³

åñáó»ë – Ñ»ñÃÇ µáÉáñ ï³ññ»ñÇ Ñ»é³óáõÙ

»Éù – ãÏ³

í»ñçÝ³å³ÛÙ³Ý – ¹³ï³ñÏ Ñ»ñÃ

í»ñç ADT Queue

úñÇÝ³Ï.

ÜÏ³ñ³·ñ»Ýù Ñ»ñÃÇ ÷á÷áËáõÃÛáõÝÁ Ýßí³Í ·áñÍáÕáõÃÛáõÝÝ»ñÇ Çñ³Ï³Ý³óáõÙÇó

Ñ»ïá (ÝÏ. 4.1):

·áñÍáÕáõÃÛáõÝ Ñ»ñÃÇ íÇ×³Ï Empty() ýáõÝÏóÇ³ÛÇ ³ñÅ»ùÁ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Front True

Rear

Insert(A)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A |  |  |  |  |

Front Rear False

Insert(B)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B |  |  |  |

Front Rear False

Delete()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B |  |  |  |  |

Front Rear False

ÜÏ. 4.1 Ð»ñÃÇ íÇ×³ÏÁ Ýßí³Í ·áñÍáÕáõÃÛáõÝÝ»ñÇó Ñ»ïá

ºñÏÏáÕÙ³ÝÇ Ñ»ñÃÁ (¹»Ï, DeQueue) ³ÛÝåÇëÇ ·Í³ÛÇÝ óáõó³Ï ¿, áñáõÙ ï³ññ»ñÇ ¨ ³í»É³óÙ³Ý, ¨ Ñ»é³óÙ³Ý ·áñÍáÕáõÃÛáõÝÝ»ñÁ Ï³ï³ñíáõÙ »Ý »ñÏáõ Í³Ûñ»ñÇó: ú·ï³·áñÍáõÙ »Ý Ý³¨ ¹»ÏÇ Ù³ëÝ³íáñ ¹»åù»ñ` ¹»Ï ë³ÑÙ³Ý³÷³Ï Ùáõïùáí (ïíÛ³ÉÝ»ñÇ ³í»É³óáõÙÁ Ï³ï³ñíáõÙ ¿ ÙÇ³ÛÝ ÙÇ Í³ÛñÇó) ¨ ¹»Ï ë³ÑÙ³Ý³÷³Ï »Éùáí (ïíÛ³ÉÝ»ñÇ Ñ»é³óáõÙÁ Ï³ï³ñíáõÙ ¿ ÙÇ³ÛÝ ÙÇ Í³ÛñÇó):

4.2 Ð»ñÃÇ Çñ³Ï³Ý³óáõÙ

Ð»ñÃÁ, ÇÝãå»ë ¨ ·Í³ÛÇÝ óáõó³ÏÁ, Ï³ñ»ÉÇ ¿ Çñ³Ï³Ý³óÝ»É »ñÏáõ »Õ³Ý³Ïáí` ½³Ý·í³ÍÇ ¨ Ï³å³Ïóí³Í óáõó³ÏÇ ÙÇçáóáí: ¼³Ý·í³ÍÇ ÙÇçáóáí Ñ»ñÃÇ Çñ³Ï³Ý³óáõÙÁ ³ñ¹ÛáõÝ³í»ï ã¿, ù³ÝÇ áñ Ñ»ñÃÇó ï³ññ Ñ»é³óÝ»ÉÇë, »Ã» Ñ»ñÃÇ ëÏ½µÇ óáõóÇãÁ ÙÝáõÙ ¿ ³Ý÷á÷áË, ³å³ å³Ñ³ÝçíáõÙ ¿ ÙÝ³ó³Í ï³ññ»ñÇ ï»Õ³ß³ñÅ:

ºÃ» Ñ»ñÃÇ ëÏ½µÇ óáõóÇãÁ ï³ññ Ñ»é³óÝ»ÉÇë ÷á÷áËíáõÙ ¿, ³å³ Ñ»ñÃÇ Ñ»ï ³ßË³ï»ÉÇë Ï³ñáÕ ¿ ëï»ÕÍí»É ·»ñÑ³·»óÙ³Ý Çñ³íÇ×³Ï, ÙÇÝã¹»é Ñ»ñÃÝ Çñ³Ï³Ý³óÝáÕ ½³Ý·í³ÍÇ ëÏ½µáõÙ Ï³ñáÕ »Ý ÉÇÝ»É ³½³ï ¹Çñù»ñ:

Ð»ñÃÇ ³ñ¹ÛáõÝ³í»ï Çñ³Ï³Ý³óáõÙ Ï³ñ»ÉÇ ¿ ³å³Ñáí»É óÇÏÉÇÏ ½³Ý·í³ÍÇ û·ÝáõÃÛ³Ùµ: ²Ûë ¹»åùáõÙ ï³ññ»ñÇ Ñ»é³óÙ³Ý ³ñ¹ÛáõÝùáõÙ ÙÝ³ó³Í ï³ññ»ñÇ ï»Õ³ß³ñÅ ãÇ Ï³ï³ñíáõÙ ¨ ³½³ïí³Í ¹Çñù»ñÇ ÏñÏÇÝ û·ï³·áñÍÙ³Ý ßÝáñÑÇí áñáß ÇÙ³ëïáí ÉáõÍíáõÙ ¿ Ñ»ñÃÇ ·»ñÑ³·»óÙ³Ý Ñ³ñóÁ: î³ññÇ Ñ»é³óÙ³Ý/³í»É³óÙ³Ý ¹»åùáõÙ front ¨ rear ÷á÷áË³Ï³ÝÝ»ñÁ ÷á÷áËíáõÙ »Ý óÇÏÉÇÏ Ó¨áí:

Ð»ñÃÇ Çñ³Ï³Ý³óáõÙÁ Ï³å³Ïóí³Í óáõó³ÏÇ ÙÇçáóáí

²Ûë ¹»åùáõÙ Ñ»ñÃÁ Ï³½Ùí³Í ¿ ³é³ÝÓÇÝ Ñ³Ý·áõÛóÝ»ñÇó ¨ áõÝÇ »ñÏáõ óáõóÇã` F ¨ R, áñáÝù Ñ³Ù³å³ï³ëË³Ý³µ³ñ ÝßáõÙ »Ý ³é³çÇÝ ¨ í»ñçÇÝ ï³ññ»ñÁ:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | . . . . |  |  |

F R

ÜÏ³ñ³·ñ»Ýù Ñ»ñÃÇ Ñ»ï Ï³ï³ñíáÕ áñáß ·áñÍáÕáõÃÛáõÝÝ»ñ:

1. Empty() - ëïáõ·»É Ñ»ñÃÁ ¹³ï³ñÏ ¿, Ã» áã:

»Ã» (F =NULL)

³å³ TRUE

³ÛÉ³å»ë FALSE

1. Insert(item) - item ï³ññÝ ³í»É³óÝ»É Ñ»ñÃÇÝ:

x ⇐ AVAIL, INFO(x) ← item, LINK(x) ← NULL

»Ã» ( F ≠ NULL)

³å³ { LINK(R) ← x, R ← x}

³ÛÉ³å»ë {R ← x, F ← x}

1. Delete() - Ñ»ñÃÇó ï³ññ Ñ»é³óÝ»É:

»Ã» (F = NULL)

³å³ UNDERFLOW

³ÛÉ³å»ë {x ← F, F ← LINK(F),

»Ã» (F = NULL) ³å³ R ← NULL,

x ⇒ AVAIL}

1. ClearQueue() - Ñ»ñÃÁ ¹³ï³ñÏ»É:

»Ã» (F = NULL)

³å³ NOP

³ÛÉ³å»ë { LINK(R) ← AVAIL, AVAIL ← F, F ← NULL, R ← NULL }

Ð»ñÃÇ Çñ³Ï³Ý³óáõÙÁ í»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛóáí Ï³å³Ïóí³Í óáõó³ÏÇ ÙÇçáóáí

Ð»ñÃÁ Ý»ñÏ³Û³óÝáÕ Ï³å³Ïóí³Í óáõó³ÏÁ Ï³ñáÕ ¿ áõÝ»Ý³É í»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛó (Head):

Head 1 Ñ³Ý·áõÛó

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | . . . . |  |  |

F R

²ÛëåÇëÇ Ñ»ñÃ»ñáõÙ í»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛóÇ INFO ¹³ßïáõÙ ·ñí³Í ÇÝýáñÙ³óÇ³Ý ³Ýï»ëíáõÙ ¿, ÇëÏ ¹³ï³ñÏ Ñ»ñÃÁ Ï³½Ùí³Í ¿ ÉÇÝáõÙ ÙÇ³ÛÝ í»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛóÇó, áñÇ LINK ¹³ßïáõÙ ·ñí³Í ¿ NULL:

ì»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛóáí Ñ»ñÃÇ Ñ»ï Ï³ï³ñíáÕ ·áñÍáÕáõÃÛáõÝÝ»ñÝ áõÝ»Ý Ñ»ï¨Û³É Çñ³Ï³Ý³óáõÙÝ»ñÁ:

1. Empty() - ëïáõ·»É Ñ»ñÃÁ ¹³ï³ñÏ ¿, Ã» áã

»Ã» (F = R )

³å³ TRUE

³ÛÉ³å»ë FALSE

1. Insert(item) - item ï³ññÝ ³í»É³óÝ»É Ñ»ñÃÇÝ

x ⇐ AVAIL, INFO(x) ← item, LINK(x) ← NULL, LINK(R) ← x, R ← x

1. Delete() - Ñ»ñÃÇó ï³ññ Ñ»é³óÝ»É

»Ã» (F = R)

³å³ UNDERFLOW

³ÛÉ³å»ë {x←LINK(F), LINK(F)←LINK(x),

»Ã» (x = R) ³å³ { R←F}

x⇒AVAIL}

1. ClearQueue() - Ñ»ñÃÁ ¹³ï³ñÏ»É

»Ã» (F = R)

³å³ NOP

³ÛÉ³å»ë { LINK(R) ← AVAIL, AVAIL ← LINK(F), LINK(F) ← NULL, R ← F}

4.3 Ð»ñÃÇ Çñ³Ï³Ý³óáõÙÁ C++ É»½íáí

Æñ³Ï³Ý³óÝ»Ýù Ñ»ñÃÁ C++ É»½íáí Áëï Ñ»ñÃÇ ßñç³Ý³Ó¨ Ùá¹»ÉÇ ¨ í»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛóáí ·Í³ÛÇÝ óáõó³ÏÇ:

Ð»ñÃÇ Çñ³Ï³Ý³óáõÙÁ ßñç³Ý³Ó¨ Ùá¹»ÉáõÙ

Þñç³Ý³Ó¨ Ùá¹»ÉáõÙ Ñ»ñÃÁ Çñ³Ï³Ý³óíáõÙ ¿ óÇÏÉÇÏ ½³Ý·í³ÍÇ ÙÇçáóáí: ¼³Ý·í³ÍÇ ï³ññ»ñÇ ù³Ý³ÏÁ ïñíáõÙ ¿ MaxQSize Ñ³ëï³ïáõÝáí: front-Á (0<=front<= MaxQSize) óáõÛó ¿ ï³ÉÇë Ñ»ñÃÇ ³é³çÇÝ ï³ññÇ ¹ÇñùÁ ¨ ï³ññÇ Ñ»é³óÙ³Ý ¹»åùáõÙ ³ÛÝ ß³ñÅíáõÙ ¿ ßñç³Ý³·Íáí: rear-Á (0<=rear<= MaxQSize) óáõÛó ¿ ï³ÉÇë Ñ»ñÃÇ Ù»ç ï³ññ ³í»É³óÝ»Éáõ ¹ÇñùÁ: î³ññ ³í»É³óÝ»Éáõó Ñ»ïá rear-Á ÝáõÛÝå»ë ß³ñÅíáõÙ ¿ ßñç³Ý³·Íáí: count-Á Ñ»ñÃÇ ï³ññ»ñÇ ù³Ý³ÏÝ ¿: ºñµ count=MaxQSize` Ñ»ñÃÁ ÉÇùÝ ¿:

rear ¨ front ÷á÷áË³Ï³ÝÝ»ñÁ ÷áËíáõÙ »Ý Ñ»ï¨Û³É Ï»ñå`

rear = (rear+1) % MaxQSize, front = (front+1) % MaxQSize:

MaxQSize=4; count=3

rear front

A

B

C

MaxQSize=4; count=3

rear

D

B

C front

³í»É³óÝ»É D-Ý

MaxQSize=4; count=2

rear

B

C front

Ñ»é³óÝ»É A-Ý

MaxQSize=4; count=4

D E

B rear

C front

³í»É³óÝ»É E-Ý

MaxQSize=4; count=3

D E

front C rear

Ñ»é³óÝ»É B-Ý

ÜÏ. 4.2 Ð»ñÃÇ ßñç³Ý³Ó¨ Ùá¹»É

Î³½Ù»Ýù Queue ¹³ëÁ, áñÝ Çñ³Ï³Ý³óÝáõÙ ¿ Ñ»ñÃÁ û·ï³·áñÍ»Éáí qlist óÇÏÉÇÏ ½³Ý·í³Í: front, rear ÷á÷áË³Ï³ÝÝ»ñÁ áñáßáõÙ »Ý Ñ»ñÃÇ ëÏÇ½µÁ ¨ í»ñçÁ, ÇëÏ count-Á` Ñ»ñÃÇ ï³ññ»ñÇ ù³Ý³ÏÁ: DataType å³ñ³Ù»ïñÇ½³óí³Í ïÇåÁ ÑÝ³ñ³íáñáõÃÛáõÝ ¿ ï³ÉÇë Ñ»ñÃÁ Çñ³Ï³Ý³óÝ»É ï³ñµ»ñ ïÇåÇ ïíÛ³ÉÝ»ñÇ Ñ³Ù³ñ:

Queue ¹³ëÇ Ñ³Ûï³ñ³ñáõÙÁ Ï³ï³ñ»Ýù aqueue.Ñ ý³ÛÉáõÙ:

// Ñ»ñÃÝ Çñ³Ï³Ý³óÝáÕ ½³Ý·í³ÍÇ ã³÷Á

const int MaxQSize =50 ;

class Queue

{

private:

// ÷á÷áË³Ï³ÝÝ»ñ, ½³Ý·í³Í

int front, rear, count ;

DataType qlist[MaxQSize] ;

public :

// ÏáÝëïñáõÏïáñ

Queue() ;

// Ñ»ñÃÇ ÷á÷áËÙ³Ý ·áñÍáÕáõÃÛáõÝÝ»ñ

void QInsert (const DataType & item) ;

DataType QDelete();

void ClearQ();

// Ñ»ñÃÇ Ñ³ë³Ý»ÉÇáõÃÛ³Ý ·áñÍáÕáõÃÛáõÝ

DataType QFront() const;

// Ñ»ñÃÇ íÇ×³ÏÇ ëïáõ·Ù³Ý ·áñÍáÕáõÃÛáõÝÝ»ñ

int QEmpty() const ;

int QFull() const ;

int QLength() const ;

};

Queue ¹³ëÇ Çñ³Ï³Ý³óáõÙÁ Ï³ï³ñ»Ýù aqueue.cpp ý³ÛÉáõÙ:

//Queue() ÏáÝëïñáõÏïáñÁ ëï»ÕÍáõÙ ¿ ¹³ï³ñÏ Ñ»ñÃ:

Queue::Queue(void): front(0), rear(0), count(0)

{ }

//QInsert()-Á DataType ïÇåÇ item ï³ññÝ ³í»É³óÝáõÙ ¿ Ñ»ñÃÇ í»ñçáõÙ:

void Queue::QInsert (const DataType & item)

{ // »Ã» Ñ»ñÃÁ ÉÇùÝ ¿ ³í³ñï»É Íñ³·ÇñÁ

if (count==MaxQSize)

{

cerr<< “Ð»ñÃÁ Ñ³·»óí³Í ¿“<< endl;

exit(1);

}

// ³í»É³óÝ»É count-Á, ½³Ý·í³ÍáõÙ ·ñ»É item-Á, ÷áË»É rear-Á

count++;

qlist[rear]=item;

rear=(rear+1)%MaxQSize;

}

//QDelete()-Á Ñ»ñÃÇó Ñ»é³óÝáõÙ ¿ ï³ññ ¨ í»ñ³¹³ñÓÝáõÙ ¿ Ýñ³ ³ñÅ»ùÁ:

DataType Queue::QDelete()

{

DataType temp;

// »Ã» Ñ»ñÃÁ ¹³ï³ñÏ ¿ ³í³ñï»É Íñ³·ÇñÁ

if(count == 0)

{

cerr <<”Ð»é³óáõÙ ¹³ï³ñÏ Ñ»ñÃÇó” <<endl;

exit(1);

}

// ÷áùñ³óÝ»É count-Á, ï»Õ³ß³ñÅ»É Ñ»ñÃÇ ëÏÇ½µÁ ¨ Ñ»é³óí³Í ï³ññÁ í»ñ³¹³ñÓÝ»É

temp=qlist[front];

count--;

front=(front+1)% MaxQSize;

return temp;

}

ì»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛóáí Ñ»ñÃÇ Çñ³Ï³Ý³óáõÙÁ

²ÛÅÙ Ï³éáõó»Ýù Queue ¹³ëÇ ß³µÉáÝÁ, áñÁ Ùá¹»É³íáñáõÙ ¿ Ï³å³Ïóí³Í ·Í³ÛÇÝ óáõó³Ïáí Çñ³Ï³Ý³óí³Í í»ñÝ³·ñ³ÛÇÝ Ñ³Ý·áõÛóáí Ñ»ñÃ: àñå»ë ß³µÉáÝÇ å³ñ³Ù»ïñ Ñ³Ù³ñ»Ýù Ñ»ñÃÇ ï³ññ»ñÇ ïÇåÁ: ÜÏ. 4.3-áõÙ Ý»ñÏ³Û³óí³Í ¿ Queue ¹³ëÇ ß³µÉáÝÇ UML-¹Ç³·ñ³ÙÁ:



ÜÏ. 4.3

ÜÏ³ñ³·ñ»Ýù Ñ»ñÃÇ Ñ³Ý·áõÛóÝ»ñÇ Node ïÇåÁ:

template <class T>

struct Node

{ T info;

Node<T> \*link;

};

Queue ¹³ëÇ Ñ³Ûï³ñ³ñáõÙ:

template <class T>

class Queue

{ private:

Node <T> head;

Node <T> \*first;

Node <T> \*last;

public:

Queue(): first(&head), last(&head) { }

~ Queue ();

void Insert (T item);

void Delete ();

void GetFirst (T & item);

bool Empty() const;

};

Queue ¹³ëÇ Çñ³Ï³Ý³óáõÙ:

//Insert()–Á item-Ý ³í»É³óÝáõÙ ¿ Ñ»ñÃÇÝ:

template <class T>

void Queue <T>::Insert (T item)

{ Node <T> \*p = new Node<T>;

If (!p) throw “Overflow I heap”;

p→info = item;

p→link = 0;

last→link = p;

last = p;

}

//Delete()-Á Ñ»ñÃÇó Ñ»é³óÝáõÙ ¿ ï³ññ:

template <class T>

void Queue <T>::Delete()

{ if (Empty()) throw ”Empty Queue”;

Node <T> \*p = head.link;

head.link = p→link;

delete p;

if (head.link = = 0)

{ last =&head; head.link=NULL; }

return;

}

//Empty()–Á ëïáõ·áõÙ ¿ Ñ»ñÃÇ ¹³ï³ñÏ ÉÇÝ»ÉÁ:

template <class T>

bool Queue <T>:: Empty ()const

{

return last == &head;

}

//GetFirst()-Á í»ñ³¹³ñÓÝáõÙ ¿ Ñ»ñÃÇ ³é³çÇÝ ï³ññÇ ³ñÅ»ùÁ:

template <class T>

void Queue <T>::GetFirst (T &item)

{ if (Empty()) throw ”Empty Queue”;

item = head.link→info;

return;

}

//~ Queue() ¹»ëïñáõÏïáñÝ ³½³ïáõÙ ¿ Ñ»ñÃÇ ½µ³Õ»óñ³Í ïÇñáõÛÃÁ:

template <class T>

Queue <T>:: ~ Queue()

{ Node <T> \*p;

if (first != last)

{ p = first→link;

while (p)

{ first→link = p→link;

delete (p);

p = first→link;

}

last = first;

}

}

´»ñ»Ýù Queue ¹³ëÇ ß³µÉáÝÝ û·ï³·áñÍáÕ Íñ³·ñÇ ûñÇÝ³Ï:

# include <iostream.h>

# include “Queue.h”

int main()

{ Queue <int> IQ; // ëï»ÕÍíáõÙ ¿ int ïÇåÇ ï³ññ»ñÇó Ï³½Ùí³Í Ñ»ñÃ

Queue <char\*> CQ; // ëï»ÕÍíáõÙ ¿ óáõóÇãÝ»ñ å³ñáõÝ³ÏáÕ ï³ññ»ñÇ Ñ»ñÃ

IQ.Insert(30); // 30-Á ³í»É³óíáõÙ ¿ Ñ»ñÃÇÝ

IQ.Insert(70); // 70-Á ³í»É³óíáõÙ ¿ Ñ»ñÃÇÝ

CQ.Insert(“Hello!”); // Hello! ïáÕÝ ³í»É³óíáõÙ ¿ Ñ»ñÃÇÝ

cout << IQ.GetFirst() << endl; // ïåáõÙ ¿ ` ³é³çÇÝ ï³ññÁ` 30

if ( ! IQ.Empty ( ) )

IQ.Delete();

cout << IQ. GetFirst() << endl; // ïåáõÙ ¿ 70

return 0;

}

4.5 Ü³Ë³å³ïíáõÃÛ³Ùµ Ñ»ñÃ

ÆÝãå»ë ·Çï»Ýù Ñ»ñÃÁ ïíÛ³ÉÝ»ñÇ Ï³éáõóí³Íù ¿, áñÝ ³å³ÑáíáõÙ ¿ ï³ññ»ñÇ FIFO Ï³ñ·: î³ññ»ñÁ h»ñÃÇó Ñ»é³óíáõÙ »Ý ëÏë³Í Ñ»ñÃÇ ëÏ½µáõÙ ¹ñí³Í ï³ññÇó: ê³Ï³ÛÝ, Ñ³×³Ë Ñ³ñÏ ¿ ÉÇÝáõÙ û·ï³·áñÍ»É ³ÛÝåÇëÇ Ñ»ñÃ, áñÇ ï³ññ»ñÁ ûÅïí³Í »Ý Ý³Ë³å³ïíáõÃÛ³Ùµ ¨ Ñ»ñÃ³Ï³Ý Ñ»é³óíáÕ ï³ññÝ áõÝÇ ³Ù»Ý³µ³ñÓñ Ý³Ë³å³ïíáõÃÛáõÝÁ: ²Û¹åÇëÇ Ï³éáõóí³ÍùÁ ÏáãíáõÙ ¿ Ý³Ë³å³ïíáõÃÛ³Ùµ Ñ»ñÃ (priority queue):

Pqueue ¹³ë

²Ûëï»Õ Ý³Ë³å³ïíáõÃÛ³Ùµ Ñ»ñÃÝ ÏÇñ³Ï³Ý³óí³Í ¿ ½³Ý·í³ÍÇ ÙÇçáóáí, áñÇ ï³ññ»ñÁ Datatype ïÇåÇ »Ý:

Pqueue ¹³ëÇ Ñ³Ûï³ñ³ñáõÙ:

// Ý³Ë³å³ïíáõÃÛ³Ùµ Ñ»ñÃÇ Ù³ùëÇÙ³É ã³÷Á

const int MaxPQSize = 50 ;

class Pqueue

{

private:

// Ñ»ñÃÁ Ý»ñÏ³Û³óÝáÕ ½³Ý·í³ÍÁ ¨ Ñ»ñÃÇ ï³ññ»ñÇ ù³Ý³ÏÁ

DataType pqlist [MaxPQSize] ;

int count ;

public :

// ÏáÝëïñáõÏïáñ

Pqueue () ;

//Ñ»ñÃÇ ÷á÷áËÙ³Ý ·áñÍáÕáõÃÛáõÝÝ»ñ

void PQInsert (const DataType & item ) ;

DataType PQDelete();

void ClearPQ();

//Ñ»ñÃÇ íÇ×³ÏÇ ëïáõ·Ù³Ý ·áñÍáÕáõÃÛáõÝÝ»ñ

int PQEmpty() const ;

int PQFull(void) const ;

int PQLength(void) const ;

};

´»ñ»Ýù Pqueue ¹³ëÝ û·ï³·áñÍáÕ Íñ³·ñÇ Ñ³ïí³Í: ºÝÃ³¹ñ»Ýù µ³ñÓñ Ý³Ë³å³ïíáõÃÛáõÝ áõÝ»óáÕ ï³ññÁ ¹³ ³Ù»Ý³÷áùñ ³ñÅ»ù áõÝ»óáÕ ï³ññÝ ¿:

typedef int DataType; //Ñ³Ûï³ñ³ñíáõÙ ¿ Ñ»ñÃÇ ï³ññ»ñÇ ïÇåÁ

Pqueue PQ; //ëï»ÕÍíáõÙ ¿ ¹³ï³ñÏ Ñ»ñÃ

PQ. PQInsert (20); //Ñ»ñÃÇÝ ³í»É³óíáõÙ ¿ 20

PQ.PQInsert (10); //Ñ»ñÃÇÝ ³í»É³óíáõÙ ¿ 10

cout <<PQ.PQLength () << endl ; // ïåíáõÙ ¿ 2

N =PQ.PQDelete () ; //10-Á Ñ³ÝíáõÙ ¿ Ñ»ñÃÇó ¨ í»ñ³·ñíáõÙ N-ÇÝ

#### Pqueue ¹³ëÇ Çñ³Ï³Ý³óáõÙÁ

#### Pqueue ¹³ëÇ Çñ³Ï³Ý³óáõÙÁ ÝÙ³Ý ¿ Queue ¹³ëÇ Çñ³Ï³Ý³óÙ³ÝÁ` µ³ó³éáõÃÛ³Ùµ PQDelete ·áñÍáÕáõÃÛ³Ý: ²Ûë Ù»Ãá¹Á Ñ»ñÃÇó Ñ»é³óÝáõÙ ¿ ³Ù»Ý³µ³ñÓñ Ý³Ë³å³ïíáõÃÛ³Ùµ ï³ññÁ: Ø»ñ ¹»åùáõÙ ¹³ ³Ù»Ý³÷áùñ ³ñÅ»ù áõÝ»óáÕ ï³ññÝ ¿:

#### PQDelete Ñ»ñÃÇó Ñ»é³óÝáõÙ ¿ ³Ù»Ý³÷áùñ ³ñÅ»ù áõÝ»óáÕ ï³ññÁ ¨ í»ñ³¹³ñÓÝáõÙ Ýñ³ ³ñÅ»ùÁ:

DataType PQueue::PQDelete()

{

DataType min ;

int i, minindex = 0;

if ( count > 0 )

{

// pqlist ½³Ý·í³ÍáõÙ ·ïÝ»É ÷áùñ³·áõÛÝ ³ñÅ»ùÁ ¨ ¹ñ³ ÇÝ¹»ùëÁ

min = pqlist [0] ;

for( i = 1; I < count; i ++)

if ( pqlist[ i ] < min )

{ min = pqlist [ i ]; minindex = i; }

// Ñ»ñÃÇ í»ñçÇÝ ï³ññÁ ï»Õ³÷áË»É ³Ù»Ý³÷áùñ ï³ññÇ ï»ÕÁ, Ñ³ßíÇãÁ Ù»Ïáí //÷áùñ³óÝ»É

pqlist[minindex] = pqlist[count -1];

count -- ;

}

// »Ã» pqlist ½³Ý·í³ÍÁ ¹³ï³ñÏ ¿, ³å³ ³í³ñï»É Íñ³·ÇñÁ

else

{

cerr << “Ñ»é³óáõÙ ¹³ï³ñÏ Ñ»ñÃÇó” << endl ;

exit (1) ;

}

return min ;

}

ÊÝ¹ÇñÝ»ñ

1. \* Æñ³Ï³Ý³óÝ»É Ñ»ñÃÁ óÇÏÉÇÏ ½³Ý·í³ÍÇ ÙÇçáóáí:
2. Æñ³Ï³Ý³óÝ»É Ñ»ñÃÇ ßñç³Ý³Ó¨ Ùá¹»ÉÁ ³é³Ýó Ñ»ñÃÇ ï³ññ»ñÇ ù³Ý³ÏÇ ÷á÷áË³Ï³ÝÇ û·ï³·áñÍÙ³Ý: Ü³Ë³ï»ë»É Ù»Ï “¹³ï³ñÏ” ï³ññ Ñ»ñÃÇ ëÏ½µáõÙ: Ü³Ë³ë»ë»É µ³ó³éÇÏ Çñ³íÇ×³ÏÝ»ñÇ Ùß³ÏáõÙ:
3. \* ØÇ³ÝÇß ¨ »ñÏÝÇß µÝ³Ï³Ý Ãí»ñÇó Ï³½Ùí³Í Ñ³çáñ¹³Ï³ÝáõÃÛáõÝÁ ¹³ë³íáñ»É ãÝí³½Ù³Ý Ï³ñ·áí é³¹Çùë (radix) Ï³ñ·³íáñÙ³Ý ÙÇçáóáí:
4. è³¹Çùë Ï³ñ·³íáñáõÙÁ Çñ³Ï³Ý³óÝ»É Ï³Ù³Û³Ï³Ý µÝ³Ï³Ý Ãí»ñÇ Ñ³çáñ¹³Ï³ÝáõÃÛ³Ý Ñ³Ù³ñ:
5. îñí³Í »Ý Q1 ¨ Q2 Ñ»ñÃ»ñÁ: Î³éáõó»É Q3 Ñ»ñÃÁ, áñÇ ³é³çÇÝ Ù³ëáõÙ Q1 -Ç ï³ññ»ñÝ »Ý, ÇëÏ »ñÏñáñ¹ Ù³ëáõÙ` Q2-Ç ï³ññ»ñÁ ßñçí³Í Ï³ñ·áí: