**Pintos Project 0-2: Pintos Data Structure**

|  |  |
| --- | --- |
| 담당 교수 : | 박성용 교수님 |
| 학번 : | 20171101 |
| 이름 : | 박근영 |
|  |  |

**반드시 아래의 양식과 순서를 따라서 작성하기 바랍니다.**

1. **Additional Implementation**

|  |  |
| --- | --- |
| **Prototype** | void processCommand(char \*cmd) |
| **Parameter** | (char \*) command |
| **Return** | none |
| **Function** | 입력받은 커맨드 한 줄을 공백 기준으로 잘라 argument 배열에 저장 |

|  |  |
| --- | --- |
| **Prototype** | bool stringToBool(char \*s) |
| **Parameter** | (char \*) boolean 값을 나타내는 문자열(true or false) |
| **Return** | (bool) 문자열이 나타내는 bool 변수 |
| **Function** | 문자열을 bool 변수로 변환해 리턴 |

|  |  |
| --- | --- |
| **Prototype** | void createHandler() |
| **Parameter** | none |
| **Return** | none |
| **Function** | argument(전역변수)가 create를 가리킬 때, 입력받은 자료구조에 해당하는 함수를 CREATE flag와 함께 호출 |

|  |  |
| --- | --- |
| **Prototype** | void deleteHandler() |
| **Parameter** | none |
| **Return** | none |
| **Function** | argument(전역변수)가 delete를 가리킬 때, 입력받은 자료구조에 해당하는 함수를 DELETE flag와 함께 호출 |

|  |  |
| --- | --- |
| **Prototype** | void dumpdataHandler() |
| **Parameter** | none |
| **Return** | none |
| **Function** | argument(전역변수)가 dumpdata를 가리킬 때, 입력받은 자료구조에 해당하는 함수를 DUMP flag와 함께 호출 |

|  |  |
| --- | --- |
| **Prototype** | void listFunctions(int flag) |
| **Parameter** | (int) flag[CREATE, DELETE, DUMP, OTHERS] |
| **Return** | none |
| **Function** | list 자료구조와 관련된 다양한 기능을 수행 |

|  |  |
| --- | --- |
| **Prototype** | void hashFunctions(int flag) |
| **Parameter** | (int) flag[CREATE, DELETE, DUMP, OTHERS] |
| **Return** | none |
| **Function** | hashtable 자료구조와 관련된 다양한 기능을 수행 |

|  |  |
| --- | --- |
| **Prototype** | void bitmapFunctions(int flag) |
| **Parameter** | (int) flag[CREATE, DELETE, DUMP, OTHERS] |
| **Return** | none |
| **Function** | bitmap 자료구조와 관련된 다양한 기능을 수행 |

1. **List**

|  |  |
| --- | --- |
| **Prototype** | void list\_swap(struct list\_elem \*a, struct list\_elem \*b) |
| **Parameter** | (struct list\_elem \*) swap할 원소 a, b |
| **Return** | none |
| **Function** | Swap two list elements in parameters. |

|  |  |
| --- | --- |
| **Prototype** | void list\_shuffle(struct list \*list) |
| **Parameter** | (struct list \*) 원소를 섞을 list |
| **Return** | none |
| **Function** | Shuffle elements of LIST in the parameter |

|  |  |
| --- | --- |
| **Prototype** | struct list\_elem \*list\_find(struct list \*list, int index) |
| **Parameter** | (struct list \*) 탐색 대상 list  (index) 찾을 원소의 index |
| **Return** | (struct list\_elem \*) 해당 index의 원소 |
| **Function** | Find and return elements of LIST by index |

|  |  |
| --- | --- |
| **Prototype** | bool list\_less\_function (const struct list\_elem \*a, const struct list\_elem \*b, void \*aux); |
| **Parameter** | (const struct list\_elem \*) list element a, b to compare  (void \*) auxiliary data |
| **Return** | (bool) true(if A<B) or false(if A>=B) |
| **Function** | Compares the value of two list elements A and B, given auxiliary data AUX. Then returns bool value. |

1. **Hash Table**

|  |  |
| --- | --- |
| **Prototype** | unsigned hash\_int\_2 (int) |
| **Parameter** | (int) integer that will be hashed |
| **Return** | (unsigned int) hash value of integer I |
| **Function** | returns a hash of interger i. (32 bit mix function 사용)  출처: https://gist.github.com/badboy/6267743 |

|  |  |
| --- | --- |
| **Prototype** | unsigned hash\_hash\_function (const struct hash\_elem \*e, void \*aux) |
| **Parameter** | (const struct hash\_elem\*) hash element E  (void\*) auxiliary data |
| **Return** | (unsigned int) hash value |
| **Function** | Computes and returns the hash value for hash element E, given auxiliary data AUX. Uses hash\_int function. |

|  |  |
| --- | --- |
| **Prototype** | bool hash\_less\_function (const struct hash\_elem \*a, const struct hash\_elem \*b, void \*aux) |
| **Parameter** | (const struct hash\_elem\*) hash elements a, b to compare  (void\*) auxiliary data |
| **Return** | (bool) true(if A<B) or false(if A>=B) |
| **Function** | Compares the value of two hash elements A and B, given auxiliary data AUX. Then returns bool value. |

|  |  |
| --- | --- |
| **Prototype** | void hash\_destructor (struct hash\_elem \*e, void \*aux) |
| **Parameter** | (struct hash\_elem\*) hash element E  (void\*) auxiliary data |
| **Return** | none |
| **Function** | Destruct and free hash element E |

|  |  |
| --- | --- |
| **Prototype** | void hash\_multiply (struct hash\_elem \*e, void \*aux) |
| **Parameter** | (struct hash\_elem\*) hash element E  (void\*) auxiliary data (2: square, 3: triple) |
| **Return** | none |
| **Function** | 입력받은 hash element E의 data값에 대하여, AUX(flag)값에 따라 square 또는 triple 연산을 수행 |

1. **Bitmap**

|  |  |
| --- | --- |
| **Prototype** | struct bitmap \*bitmap\_expand(struct bitmap \*, int size) |
| **Parameter** | (struct bitmap\*) bitmap that you want to expand  (int) 증가시킬 사이즈 |
| **Return** | (struct bitmap\*) expanded bitmap (NULL if fail) |
| **Function** | Expand size of given bitmap |