Lab 04 (8%)
Topics: ADT Set, Hash Tables, pointers to functions

Problem 01 (2%)

Write C++ program using standard class unordered_set which reads and executes commands of following types:
+ word (add word to set s);
- word (delete word from set s);
? word (print “YES” if set s has this word, otherwise print “NO”);
(print current contexts of set s);
Word in all commands is a sequence of lowercase English letters.
Set s is a collection of unique strings. You have to use class unordered_set to represent this set in your program. Give performance characteristics of +, -, ? operations.
Use class unordered_set<string> with your own hash function which returns 42 as a hash value for any strings. Give performance characteristics in this case.

Problem 02 (4%)

Create class HashSetStr to store unique strings and use it instead of unordered_set to solve Problem 01. Class HashSetStr has to have following interface:

Constructor
HashSetStr(HashFunc hf)

Destructor
~HashSetStr()

bool insert(const string& s)
Inserts string s in hash set. Returns true if insertion was successful, otherwise returns false.

bool erase(const string& s)
Deletes string s from hash set. Returns true if deletion was successful, otherwise returns false.

bool find(const string& s) const
Searches for string s in hash set. Return true if s was found, otherwise returns false.

void clear()
Deletes all elements of container.

size_t size() const
Returns the number of elements in container.

void print() const
Prints all chains (buckets) of hash table in following order:
index of bucket: <element> <element> …

Your class has to use “Separate chaining approach” to resolve collisions of elements.
In case of command “^” you have to call method print of your hash set.

Problem 3 (1%)

The purpose of this problem is to show how we can use pointers to functions to generalize solutions of computational tasks. You have to output tables of library functions abs(x), sqrt(x) and your function sqr(x) for arbitrary ranges of x. You have to do this using only one function:

```
void printTable (
    double (*f)(double),
    const string& msg, double a, double b, double step)
```

Please, note that the first parameter is a pointer to a function values of which we want to print.

Problem 04 (1%)

The purpose of this problem is to show the importance of pointers to functions in GUI libraries. You have to write a console application which creates three objects of class Button with text “Main Button”, “Left Button”, “Right Button” and asks for pairs of coordinates (“clicks”). If a point with entered coordinates is inside of some of these buttons program has to call the corresponding function to output one of the following messages:

If it is the “Main Button” then program outputs the message “I am the Main Button”;

If it is the one of “Left Button” or “Right Button” then program outputs the message “You clicked on button <name>” where instead of <name> program uses the corresponding name.

Class Button must have the following interface:

Constructor: Button(const string& aTxt, int aX, int aY, int aW, int aH, Action anAct), where Action is a another name of a type void (*)(Button&);

Methods:

```
bool isClicked(int ax, int ay) const;

void doOnClick();

const string&  getText() const;
```

Each object of class Button must have a pointer to a function which has to be called when user “clicks” button.