1. STACKPAC
3. #pragma once
4. #ifndef STACKPAC
5. #define STACKPAC
7. template<class T, int n>
8. class STACK
9. {
10. private:
11. T st[n];
12. int z;
14. public:
15. void clearStack()
16. {
17. z = 0;
18. }
19. bool emptyStack()
20. {
21. return (z == 0) ? true : false;
22. }
23. bool fullStack()
24. {
25. return (z == n) ? true : false;
26. }
27. void pushStack(T x)
28. {
29. st[z] = x;
30. z++;
31. }
32. T popStack()
33. {
34. z--;
35. return st[z];
36. }
37. };
38. #endif
39. Program #1
41. #include <iostream>
42. #include <string>
43. #include <cstring>
44. #include "STACKPAC.h"
46. using namespace std;
48. int main()
49. {
50. STACK<int, 7>NUM;
51. STACK<char, 5>VOW;
52. STACK<string, 12>MON;
54. NUM.clearStack();
55. VOW.clearStack();
56. MON.clearStack();
58. int n[7] = { 5,9,7,6,2,3 };
59. char c[5] = { 'e','o','a','i','u' };
60. string month[12] = {"JAN","FEB","MAR","APR","JUN","JUL","AUG","SEP","OCT","NOV","DEC"};
62. cout << "Original Arrays :" << endl;
63. for (int z = 0; z < 7; z++)
64. {
65. cout << n[z]<< " ";
66. NUM.pushStack(n[z]);
67. }
68. cout << endl;
69. for (int z = 0; z < 5; z++)
70. {
71. cout << c[z] << " ";
72. VOW.pushStack(c[z]);
73. }
74. cout << endl;
75. for (int z = 0; z < 12; z++)
76. {
77. cout << month[z] << " ";
78. MON.pushStack(month[z]);
79. }
80. cout << endl;
81. cout<<" Arrays reversed: " << endl;
82. while (!NUM.emptyStack())
83. {
84. int t = NUM.popStack();
85. cout << t << " ";
86. }
87. cout << endl;
88. while (!VOW.emptyStack())
89. {
90. char t = VOW.popStack();
91. cout << t << " ";
92. }
93. cout << endl;
94. while (!MON.emptyStack())
95. {
96. string t = MON.popStack();
97. cout << t << " ";
98. }
99. cout << endl;
101. system("pause");
102. }
103. Program #2
105. #include<iostream>
106. #include "STACKPAC.h"
107. using namespace std;
109. int main()
110. {
111. STACK<int, 7> POST;
112. POST.clearStack();
114. int a = 5, b = 7, c = 8, d = 2;
115. int x, y, r;
116. char con;
117. do {
118. char p;
119. cout << "Enter a postfix Expression: ";
120. cin >> p;
122. while (p != '$')
123. {
124. switch (p)
125. {
126. case 'a':
127. POST.pushStack(a);
128. break;
129. case'b':
130. POST.pushStack(b);
131. break;
132. case'c':
133. POST.pushStack(c);
134. break;
135. case'd':
136. POST.pushStack(d);
137. break;
138. case '\*':
139. x = POST.popStack();
140. y = POST.popStack();
141. r = y \* x;
142. POST.pushStack(r);
143. break;
144. case '/':
145. x = POST.popStack();
146. y = POST.popStack();
147. r = y / x;
148. POST.pushStack(r);
149. break;
150. case '+':
151. x = POST.popStack();
152. y = POST.popStack();
153. r = y + x;
154. POST.pushStack(r);
155. break;
156. case '-':
157. x = POST.popStack();
158. y = POST.popStack();
159. r = y - x;
160. POST.pushStack(r);
161. break;
162. }
163. cin >> p;
164. }
166. cout << "Value of the Expression " << POST.popStack() << endl;
167. cout << "Continue? y or n: ";
168. cin >> con;
169. cout << endl;
170. } while (con!='n');
171. system("pause");
172. }

175. Program #3
177. #include<iostream>
178. #include<cstring>
179. #include "STACKPAC.h"
180. using namespace std;
182. int main()
183. {
184. STACK<char, 20> LR;
185. STACK<char, 20> RL;
186. STACK<char, 20> TEMP;
188. char con;
190. do {
191. LR.clearStack();
192. RL.clearStack();
193. TEMP.clearStack();
195. cout << "Enter a sentence: ";
196. char c;
198. while (cin.get(c), c!='\n')
199. {
200. if (isalpha(c))
201. {
202. c = tolower(c);
203. RL.pushStack(c);
204. TEMP.pushStack(c);
205. }
206. }

209. while (!TEMP.emptyStack())
210. {
211. char p = TEMP.popStack();
212. LR.pushStack(p);
213. }
214. while (!RL.emptyStack())
215. {
216. char c1 = RL.popStack();
217. char c2 = LR.popStack();
218. if (c1 != c2)
219. break;
220. }
221. if (RL.emptyStack() == true)
222. cout << "This is a Palindrome" << endl;
223. else
224. cout << "This is not a Palindrome" << endl;
226. cout << "Would you like to continue? Enter y for yes and n no: ";
227. cin >> con;
228. cout << endl;
230. cin.ignore(1, '\n');
232. } while (con == 'y');
234. system("pause");
236. }