|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Description | Change code | Expected Output | Passed |
| 1 | Check constructor initialises object | Vector<int> testVec; | m\_arrayLength = 0;  m\_arraySize = 0;  m\_theArray = NULL; | Y |
| 2 | Check SetSize function | Vector<int> testVec;  testVec.SetSize(10); | m\_arrayLength = 0;  m\_arraySize = 10;  m\_theArray = NOT NULL; | Y |
| 3 | Check SetSize function with 0 | Vector<int> testVec;  testVec.SetSize(0); | ERROR MSG  m\_arrayLength = 0;  m\_arraySize = 0;  m\_theArray = NULL; | Y |
| 4 | Check copy function | Vector<int> testVec, testVec02;  testVec.SetSize(10);  testVec02.CopyVec(testVec);  testVec02.print(); | Copy successful msg  m\_arrayLength = 0;  m\_arraySize = 10;  m\_theArray = NOT NULL; | Y |
| 5 | Check contructor with size parameter | Vector<int> testVec(11);  testVec.print(); | m\_arrayLength = 0;  m\_arraySize = 11;  m\_theArray = NOT NULL; | Y |
| 6 | Check copy contructor | Vector<int> testVec(17);  Vector<int> testVec02(testVec);  testVec02.print(); | m\_arrayLength = 0;  m\_arraySize = 17;  m\_theArray = NOT NULL; | Y |
| 7 | Check AddItem adds item to end of Vector | Vector<int> testVec(3);  for(int i = 0; i < 3; i++)  testVec.AddItem(i);  testVec.print(); | m\_arrayLength = 3;  m\_arraySize = 3;  m\_theArray = NOT NULL;  0  1  2 | Y |
| 8 | Test overloaded operater [] | Vector<int> testVec(3);  for(int i = 0; i < 3; i++)  testVec.AddItem(i);  for(int i = 0; i < 3; i++)  testVec[i] = i + 10;  testVec.print(); | m\_arrayLength = 3;  m\_arraySize = 3;  m\_theArray = NOT NULL;  10  11  12 | Y |
| 9 |  |  |  |  |
| 10 | Test PushBack outside of range | Vector<int> testVec(2);  for(int i = 0; i < 4; i++)  {  if(testVec.PushBack(i))  cout << "Inside Range" << endl;  else  cout << "Outside Range" << endl;  }  testVec.print(); | Inside Range  Inside Range  Outside Range  Outside Range  m\_arrayLength = 3;  m\_arraySize = 3;  m\_theArray = NOT NULL;  0  1 | Y |
| 11 | Check GetItem | Vector<int> testVec(3);  int x;  for(int i = 0; i < 3; i++)  {  testVec.PushBack(i);  testVec.GetItem(x, i);  cout << x << endl;  } | 0  1  2 | Y |
| 12 | Test GetLength and GetSize | Vector<int> testVec(3);  for(int i = 0; i < 2; i++)  testVec.PushBack(i);  cout << testVec.GetLength() << endl;  cout << testVec.GetSize() << endl; | 2  3 | Y |
| 13 | Tested with string object | Vector<string> testVec(3);  string testString = "String 0";  string emptyStr;  for(int i = 0; i < 3; i++)  {  testVec.PushBack(testString);  testVec.GetItem(emptyStr, i);  cout << emptyStr << i << endl;  } | String 00  String 01  String 02 | Y |