Arbeitsblatt 3

# Aufgabe 1

|  |
| --- |
| #include <iostream>  using namespace std;  int main(int argc, char \*argv[]) {  char msg[10];  char \*p;  //msg = "Bonjour"; // const char[8] kann nicht char[10] zugewiesen werden  p = "Bonjour";  //msg = p; // char\* kann nicht char[10] zugewiesen werden  p = msg;  p[0] = 'H', p[1] = 'i', p[2] = '\0';  cout << "Content of array p[] is: " << p << endl;  return 0;  } |

# Aufgabe 2

|  |
| --- |
| #include <iostream>  using namespace std;  int \*fillArray(int array[], int length) {  for (int i = 0; i < length; i++) {  array[i]++;  }  return array;  }  int main(int argc, char \*argv[]) {  int data[] = {0, 1, 2, 3, 4};  int length = sizeof(data) / sizeof(int);  int \*dataptr = fillArray(data, length);  for (int &i : data) {  cout << "For: " << i << endl;  }  for (int i = 0; i < length; i++) {  cout << "Array: " << dataptr[i] << endl;  }  return 0;  } |

# Aufgabe 3

|  |
| --- |
| #include <iostream>  using namespace std;  typedef unsigned int COLORREF;  typedef unsigned char BYTE;  struct DebugeImage {  int channels;  int width;  int height;  int pitch;  BYTE \*data;  };  int main(int argc, char \*argv[]) {  const int radius = 20;  const int size = 51;  const int halfsize = (size / 2) - 1;  const int squareradius = radius \* radius;  COLORREF colorrefs[size][size];  for (int y = 0; y < size; y++) {  for (int x = 0; x < size; x++) {  int squareddistance = (x - halfsize) \* (x - halfsize) + (y - halfsize) \* (y - halfsize);  colorrefs[x][y] = (squareddistance > squareradius) ? 0xFFFFFF00 : 0xFFFF0000;  }  }  DebugeImage image;  image.channels = sizeof(COLORREF);  image.width = size;  image.height = size;  image.pitch = size \* image.channels;  image.data = reinterpret\_cast<BYTE \*>(&colorrefs);  return 0;  } |

# Aufgabe 4

|  |
| --- |
| #include <iostream>  using namespace std;  bool strcmp(char \*data1, char \*data2) {  int a = 0;  while (data1[a] != '\0') {  a++;  }  int b = 0;  while (data2[b] != '\0') {  b++;  }  return a == b;  }  int main(int argc, char \*argv[]) {  char string0[] = {'J', 'a', 'a', 'v', 's', 's', '\0'};  char string1[] = {'J', 'a', 'a', 'v', 's', '\0'};  char string2[] = "Jaavs";  char string3[] = {'J', 'a', 'a'};  cout << string0 << " == " << string1 << " = " << strcmp(string0, string1) << endl;  cout << string2 << " == " << string3 << " = " << strcmp(string2, string3) << endl;  cout << string1 << " == " << string2 << " = " << strcmp(string1, string2) << endl;  return 0;  } |