```
//************************************************************************************
//************************************************************************************
//                                                                                  **
//                        CopyString                                                **
//      funkcjonalnosci:                                                            **
//                  - Kopiowanie łańcuchów znakowych                                 **
//                  - Porównywanie łańcuchów znaków                                 **
//                  - Dołączanie jednego łańcucha znakowego do innego               **
//                  - Zmiana znaku w łańcuchu znakowym                              **
//                                                                                  **
//************************************************************************************


void CopyString(char pcSource[], char pcDestination[])
{
  unsigned char ucCharacterCounter;

  for(ucCharacterCounter = 0; '\0' != pcSource[ucCharacterCounter]; ucCharacterCounter++)
  {
    pcDestination[ucCharacterCounter] = pcSource[ucCharacterCounter];
  }
  pcDestination[ucCharacterCounter] = '\0';
}

enum Result { OK, ERROR };

enum CompResult eCompareString(char pcStr1[], char pcStr2[])
{
  unsigned char ucCharacterCounter;

  for(ucCharacterCounter = 0; ('\0' != pcStr1[ucCharacterCounter]) || ('\0' != pcStr2[ucCharacterCounter]); ucCharacterCounter++)
  {
    if (pcStr1[ucCharacterCounter] != pcStr2[ucCharacterCounter])
    {
      return DIFFERENT;
    }
  }
  return EQUAL;
}
```

```
void AppendString(char pcSourceStr[], char pcDestinationStr[])
{
  unsigned char ucPointerPosition;

  for(ucPointerPosition = 0; '\0' != pcDestinationStr[ucPointerPosition]; ucPointerPosition++) {}
  CopyString(pcSourceStr, pcDestinationStr + ucPointerPosition);
}

void ReplaceCharactersInString(char pcString[], char cOldChar, char cNewChar)
{
  unsigned char ucCharacterCounter;

  for(ucCharacterCounter = 0; '\0' != pcString[ucCharacterCounter]; ucCharacterCounter++)
  {
    if(pcString[ucCharacterCounter] == cOldChar)
    {
      pcString[ucCharacterCounter] = cNewChar;
    }
  }
}
```