**SFWRENG 2MP3 – Programming for Mechatronics Fall 2018**

|  |  |
| --- | --- |
| **Exercise 1 Solution** | **Submitted By: HARNEET SINGH, 400110275, SINGHH76@MCMASTER.CA** |
| **Question #** | **Answer** |
| **Entire Program** | #include <stdio.h>  void loadValues ();  void Subtract ();  void Print ();  void transpose ();  void rank ();  void main ()  {  int m, n, input;  int r, c;  printf("\nThis program performs arithmetic operations on two matrices (A & B):\n");  printf("Consider (m) to be number of rows and (n) to be number of columns,\n");  printf("Please enter m-th dimension for the first matrix (A): ");  scanf("%d", &m);  printf("Please enter n-th dimension for the first matrix (A): ");  scanf("%d", &n);  printf("\nSimilarly, provide the m-th dimension of the second matrix (B): ");  scanf("%d", &r);  printf("Provide the n-th dimension of the second matrix (B): ");  scanf("%d", &c);  int mat\_A[m][n]; int mat\_B[r][c];  do {  input = -1;    printf("\n\nPlease select one of the following to perform appropriate action:\n");  printf("1 - Load Values in the Matrix (A or B)\n");  printf("2 - Subtract matrices (A-B or B-A)\n");  printf("3 - Print matrix (A or B)\n");  printf("4 - Transpose matrix (A or B)\n");  printf("5 - Rank of matrix (A or B)\n");  printf("6 - Exit\n");  printf("Please enter corresponding numerical value for the selected option: ");  scanf("%d", &input);  if (input == 1) {loadValues (m, n, r, c, mat\_A, mat\_B);}  if (input == 2) {Subtract (m, n, r, c, mat\_A, mat\_B);}  if (input == 3) {Print (m, n, r, c, mat\_A, mat\_B);}  if (input == 4) {transpose (m, n, r, c, mat\_A, mat\_B);}  if (input == 5) {rank(m, n, r, c, mat\_A, mat\_B);}  if (input == 6) {puts("\nPROGRAM EXITED NOW: Program termination value (of 6) is entered"); break;}  if (input == -1) {puts("Please enter a valid integer input");}  } while (input >= 1 && input <= 6);  }  void loadValues (int m, int n, int r, int c, int mat\_A[m][n], int mat\_B[r][c]){  char mat;  puts ("This part of the program lets you populate a particular matrix");  printf("Please specify which matrix needs to be loaded (A/a for A or B/b for B): ");  scanf(" %c", &mat); //added an extra space before %c to remove any leading space  if (mat == 'A' || mat == 'a'){  int x = -1;  for (int i = 0; i < m; i++){  for (int j = 0; j < n; j++){  printf("Enter the value of A[%dx%d] element: ", (i+1), (j+1));  scanf("%d", &x);  mat\_A[i][j] = x;  }  }  }  else if (mat == 'B' || mat == 'b'){  int x = -1;  for (int i = 0; i < r; i++){  for (int j = 0; j < c; j++){  printf("Enter the value of B[%dx%d] element: ", (i+1), (j+1));  scanf("%d", &x);  mat\_B[i][j] = x;  }  }  }  else {puts("!!!Please enter a valid input");}  }  void Subtract (int m, int n, int r, int c, int mat\_A[m][n], int mat\_B[r][c]){  int num;  puts("This part of the program performs subtraction on equi-dimensional matrices:");  printf("Please enter 1 for (A-B) operation\nOR enter 2 for (B-A) operation: ");  scanf("%d", &num);  if ((m == r) && (n == c)){  if (num == 1) {  puts("Following is the result of the applied operation:");  for (int i = 0; i < m; i++) {  for (int j = 0; j < n; j++){  printf("%6d", (mat\_A[i][j] - mat\_B[i][j]));  }  printf("\n");  }  }  else if (num == 2) {  puts("Following is the result of the applied operation:");  for (int i = 0; i < m; i++) {  for (int j = 0; j < n; j++){  printf("%6d", (mat\_B[i][j] - mat\_A[i][j]));  }  printf("\n");  }  }    else {puts("!!!Invalid input");}  }  else {puts("!!!Rows and Columns of the matrices must be equal");}  }  void Print (int m, int n, int r, int c, int mat\_A[m][n], int mat\_B[r][c]){  puts ("\nThis part of the program prints a user-required matrix:");  char num;  printf("Please enter A/a to view matrix A\nOR enter B/b to view matrix B: ");  scanf(" %c", &num);  if (num == 'A' || num == 'a') {  for (int i = 0; i < m; i++) {  for (int j = 0; j < n; j++){  printf("%6d", mat\_A[i][j]);  }  printf("\n");  }  }  else if (num == 'B' || num == 'b') {  for (int i = 0; i < r; i++) {  for (int j = 0; j < c; j++){  printf("%6d", mat\_B[i][j]);  }  printf("\n");  }  }  else {puts("!!!Please enter a valid matrix");}  }  void transpose (int m, int n, int r, int c, int mat\_A[m][n], int mat\_B[r][c]) {  puts ("\nThis part of the program computes transpose of a specified matrix:");  char num;  printf("Please enter A/a to transpose matrix A\nOR enter B/b to transpose matrix B: ");  scanf(" %c", &num);  if (num == 'A' || num == 'a') {  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++){  printf("%8d", mat\_A[j][i]);  }  printf("\n");  }  }  else if (num == 'B' || num == 'b') {  for (int i = 0; i < c; i++) {  for (int j = 0; j < r; j++){  printf("%8d", mat\_B[j][i]);  }  printf("\n");  }  }  else {puts("!!!Please enter a valid character");}  }  void rank (int m, int n, int r, int c, int mat\_A[m][n], int mat\_B[r][c]) {  puts ("\nThis part of the program determines the rank of a 2x2 matrix:");  char num;  printf("Please enter A/a to determine the rank of matrix A\nOR enter B/b for matrix B: ");  scanf(" %c", &num);  if (m == 2 && n == 2 && r == 2 && c == 2){  if (num == 'A' || num == 'a'){  int determinant = (((mat\_A[0][0]) \* (mat\_A[1][1])) - ((mat\_A[0][1]) \* (mat\_A[1][0])));    if (determinant) { puts ("Rank of matrix A is 2");}    else {puts ("Rank of matrix A is 1");}  }    else if (num == 'B' || num == 'b'){  int determinant = (((mat\_B[0][0]) \* (mat\_B[1][1])) - ((mat\_B[0][1]) \* (mat\_B[1][0])));  if (determinant) { puts ("Rank of matrix B is 2");}  else {puts ("Rank of matrix B is 1");}  }  else {puts("!!!Please enter a valid digit");}  }  else {puts("!!!This matrix is not 2x2");}  } |
| **Error Free Compilation** |  |
| **Option 6, Successful Execution** |  |
| **Option 1 – A** |  |
| **Option 1 - B** |  |
| **Option 2 -A** |  |
| **Option 2 – B** |  |
| **Option 3-A** |  |
| **Option 3-B** |  |
| **Option 4-A** |  |
| **Option 4-B** |  |
| **Option 5-A** |  |
| **Option 5-B** |  |
| **Prospective Error for Option 2 (Subtraction):** In case, the user tries to subtract two matrices of different dimensions i.e. the two matrices must possess equal number of rows and columns |  |
| **Prospective Error for Option 3 (Print):**  If the user forgets to initialize the elements of one/two matrices then, the matrix elements will, automatically, be assigned garbage values during compilation |  |