**Laborator №6 : *Probleme suplimentare***

1. Să se scrie un program cu un meniu interactiv în care avem următoarele opțiuni: 0. Ieșire 1. Citirea unei matrice patratice de ordinul n. 2. Afișarea elementelor matricei 3. Afișarea elementelor de pe diagonala principală 4. Afișarea elementelor de pe diagonala secundară 5. Afișarea elementelor de deasupra diagonalei principale 6. Afișarea elementelor de sub diagonala principală 7. Afișarea elementelor de deasupra diagonalei secundare 8. Afișarea elementelor de sub diagonala secundară 9. Afișarea elementelor de pe marginile matricei 10. Afișarea elementelor din colțurile matricei 11. Afișarea matricei transpuse

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

void \_menu(void)

{

printf("\n------------------Meniu Interactiv------------------\n"

"0. Iesire\n"

"1. Citirea unei matrice patratice de ordinul n\n"

"2. Afisarea elementelor matricei\n"

"3. Afisarea elementelor de pe diagonala principala\n"

"4. Afisarea elementelor de pe diagonala secundara\n"

"5. Afisarea elementelor de deasupra diagonalei principale\n"

"6. Afisarea elementelor de sub diagonala principala\n"

"7. Afisarea elementelor de deasupra diagonalei secundare\n"

"8. Afisarea elementelor de sub diagonala secundara\n"

"9. Afisarea elementelor de pe marginile matricei\n"

"10. Afisarea elementelor din colturile matricei\n"

"11. Afisarea matricei transpuse\n\n"

"Optiunea dvs : ");

}

int main(void)

{

int opt,ok,dim,i,j,m[100][100];

while (1)

{

\_menu();

scanf("%d", &opt);

printf("\n");

if (opt == 0)

exit(0);

switch (opt)

{

case 1:

{

ok = 1;

printf("Introduceti dimensiunea/ordinul matricei patratice : ");

scanf("%d", &dim);

for (i = 0; i < dim; i++)

{

for (j = 0; j < dim; j++)

{

printf("matrice[%d][%d] : ", i,j);

scanf("%d", &m[i][j]);

}

printf("\n");

}

}

break;

case 2:

{

if (ok == 1)

{

for (i = 0; i < dim; i++)

{

for (j = 0; j < dim; j++)

{

printf("%4d", m[i][j]);

}

printf("\n");

}

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 3:

{

if (ok == 1)

{

printf("Elementele diagonalei principale : ");

for (i = 0; i < dim; i++)

printf("%d ", m[i][i]);

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 4:

{

if (ok == 1)

{

printf("Elementele diagonalei secundare : ");

for (i = 0; i < dim; i++)

printf("%d ", m[i][dim - i - 1]);

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 5:

{

if (ok == 1)

{

printf("Elementele de deasupra diagonalei principale : ");

for (i = 0; i < dim; i++)

for (j = 0; j < dim; j++)

if (j > i)

printf("%d ", m[i][j]);

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 6:

{

if (ok == 1)

{

printf("Elementele de sub diagonala principala : ");

for (i = 0; i < dim; i++)

for (j = 0; j < dim; j++)

if (i > j)

printf("%d ", m[i][j]);

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 7:

{

if (ok == 1)

{

printf("Elementele de deasupra diagonalei secundare : ");

for (i = 0; i < dim; i++)

for (j = 0; j < dim; j++)

if (i+j < dim-1)

printf("%d ", m[i][j]);

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 8:

{

if (ok == 1)

{

printf("Elementele de sub diagonala secundara : ");

for (i = 0; i < dim; i++)

for (j = 0; j < dim; j++)

if (i + j > dim - 1)

printf("%d ", m[i][j]);

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 9:

{

if (ok == 1)

{

printf("Elementele de pe marginile matricei : ");

for (i = 0; i < dim; i++)

{

if ((i == 0) || (i == dim - 1))

{

for (j = 0; j < dim; j++)

printf("%d ", m[i][j]);

}

else

printf("%d %d ", m[i][0], m[i][dim - 1]);

}

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 10:

{

if (ok == 1)

printf("Elementele din colturile matricei : %d %d %d %d", m[0][0],m[0][dim-1],m[dim-1][0],m[dim-1][dim-1]);

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 11:

{

if (ok == 1)

{

printf("Matricea transpusa : \n");

for (i = 0; i < dim; i++)

{

for (j = 0; j < dim; j++)

{

printf("%4d", m[j][i]);

}

printf("\n");

}

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

default:

printf("Eroare ! Optiune nedefinita !\n");

break;

}

}

printf("\n");

return 0;

}

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

void \_menu(void)

{

printf("\n------------------Meniu Interactiv------------------\n"

"0. Iesire\n"

"2. Afisarea elementelor matricei\n"

"3. Afisarea elementelor de pe diagonala principala\n"

"4. Afisarea elementelor de pe diagonala secundara\n"

"5. Afisarea elementelor de deasupra diagonalei principale\n"

"6. Afisarea elementelor de sub diagonala principala\n"

"7. Afisarea elementelor de deasupra diagonalei secundare\n"

"8. Afisarea elementelor de sub diagonala secundara\n"

"9. Afisarea elementelor de pe marginile matricei\n"

"10. Afisarea elementelor din colturile matricei\n"

"11. Afisarea matricei transpuse\n\n"

"Optiunea dvs : ");

}

int main(void)

{

int opt, ok=1, dim, i, j;

printf("Introduceti dimensiunea/ordinul matricei patratice : ");

scanf("%d", &dim);

int \*\*m = (int\*\*)malloc(dim \* sizeof(int));

for (i = 0; i < dim; i++)

m[i] = (int\*)malloc(dim \* sizeof(int));

for (i = 0; i < dim; i++)

{

for (j = 0; j < dim; j++)

{

printf("matrice[%d][%d] : ", i, j);

scanf("%d", (\*(m + i) + j));

}

printf("\n");

}

while (1)

{

\_menu();

scanf("%d", &opt);

printf("\n");

if (opt == 0)

exit(0);

switch (opt)

{

case 2:

{

if (ok == 1)

{

for (i = 0; i < dim; i++)

{

for (j = 0; j < dim; j++)

{

printf("%4d", \*(\*(m+i)+j));

}

printf("\n");

}

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 3:

{

if (ok == 1)

{

printf("Elementele diagonalei principale : ");

for (i = 0; i < dim; i++)

printf("%d ", \*(\*(m+i)+i));

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 4:

{

if (ok == 1)

{

printf("Elementele diagonalei secundare : ");

for (i = 0; i < dim; i++)

printf("%d ", \*(\*(m+i)+(dim-i-1)));

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 5:

{

if (ok == 1)

{

printf("Elementele de deasupra diagonalei principale : ");

for (i = 0; i < dim; i++)

for (j = 0; j < dim; j++)

if (j > i)

printf("%d ", \*(\*(m+i)+j));

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 6:

{

if (ok == 1)

{

printf("Elementele de sub diagonala principala : ");

for (i = 0; i < dim; i++)

for (j = 0; j < dim; j++)

if (i > j)

printf("%d ", \*(\*(m+i)+j));

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 7:

{

if (ok == 1)

{

printf("Elementele de deasupra diagonalei secundare : ");

for (i = 0; i < dim; i++)

for (j = 0; j < dim; j++)

if (i + j < dim - 1)

printf("%d ", \*(\*(m+i)+j));

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 8:

{

if (ok == 1)

{

printf("Elementele de sub diagonala secundara : ");

for (i = 0; i < dim; i++)

for (j = 0; j < dim; j++)

if (i + j > dim - 1)

printf("%d ", \*(\*(m+i)+j));

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 9:

{

if (ok == 1)

{

printf("Elementele de pe marginile matricei : ");

for (i = 0; i < dim; i++)

{

if ((i == 0) || (i == dim - 1))

{

for (j = 0; j < dim; j++)

printf("%d ", \*(\*(m+i)+j));

}

else

printf("%d %d ", \*(\*(m+i)+0), \*(\*(m + i) + (dim-1)));

}

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 10:

{

if (ok == 1)

printf("Elementele din colturile matricei : %d %d %d %d", \*(\*(m+0)+0), \*(\*(m+0)+(dim-1)), \*(\*(m+(dim-1))+0), \*(\*(m+(dim-1))+(dim-1)));

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 11:

{

if (ok == 1)

{

printf("Matricea transpusa : \n");

for (i = 0; i < dim; i++)

{

for (j = 0; j < dim; j++)

{

printf("%4d", \*(\*(m+j)+i));

}

printf("\n");

}

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

default:

printf("Eroare ! Optiune nedefinita !\n");

break;

}

}

free(m);

printf("\n");

return 0;

}

**2)** Să se scrie un program în C care, folosind un meniu interactiv se pot realiza următoarele: 0. Ieșire 1. Citirea unui vector cu n elemente numere întregi 2. Afișarea elementelor vectorului 3. Afișarea sumei elementelor pare ale vectorului – folosind operații pe biți 4. Afișarea produsului elementelor impare ale vectorului – folosind operații pe biți 5. Afișarea elementelor vectorului în ordine crescătoare 6. Afișarea elementelor vectorului care sunt divizibile cu un număr precizat de utilizator de la tastatură 7. Să se afișeze elementul cel mai mare al vectorului 8. Să se afișeze elementul cel mai mic al vectorului 9. Să se calculeze produsul dintre elementul pozitiv cel mai mic al vectorului cu elementul cel mai mare al vectorului. 10. Afișarea mediei aritmetice a elementelor vectorului 11. Să se adune la toate elementele pare ale vectorului un număr precizat de utilizator de la tastatură, apoi să se afișeze noul vector.  
  
#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

void \_menu(void)

{

printf("\n------------------Meniu Interactiv------------------\n"

"0. Iesire\n"

"1. Citirea unui vector cu n elemente numere intregi\n"

"2. Afisarea elementelor vectorului\n"

"3. Afisarea sumei elementelor pare ale vectorului\n"

"4. Afisarea produsului elementelor impare ale vectorului\n"

"5. Afisarea elementelor vectorului in ordine crescatoare\n"

"6. Afisarea elementelor vectorului care sunt divizibile cu un numar\n"

"7. Afisarea elementului cel mai mare din vector\n"

"8. Afisarea elementului cel mai mic\n"

"9. Calculul produsului dintre elementul pozitiv cel mai mic cu cel mai mare\n"

"10. Afisarea mediei aritmetice\n"

"11. Adunarea unui numar la toate elementele pare a vectorului, apoi afisarea noului vector\n\n"

"Optiunea dvs : ");

}

int main(void)

{

int opt, ok=0, dim, i, j, v[100],w[100], z[100], s, p, aux, nr, min, max;

while (1)

{

\_menu();

scanf("%d", &opt);

printf("\n");

if (opt == 0)

exit(0);

switch (opt)

{

case 1:

{

ok = 1;

printf("Introduceti dimensiunea/ordinul vectorului : ");

scanf("%d", &dim);

for (i = 0; i < dim; i++)

{

printf("vector[%d] : ", i);

scanf("%d", &v[i]);

w[i] = v[i];

}

printf("\n");

}

break;

case 2:

{

if (ok == 1)

{

printf("Elementele vectorului : ");

for (i = 0; i < dim; i++)

{

printf("%d ", v[i]);

}

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 3:

{

if (ok == 1)

{

s = 0;

for (i = 0; i < dim; i++)

if ((v[i] & 1) == 0)

s += v[i];

printf("Suma elementelor pare : %d\n", s);

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 4:

{

if (ok == 1)

{

p = 1;

for (i = 0; i < dim; i++)

if ((v[i] & 1) == 1)

p \*= v[i];

printf("Produsul elementelor impare : %d\n", p);

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 5:

{

if (ok == 1)

{

for (i = 0; i < dim-1; i++)

for (j = 0; j < dim-i-1; j++)

if (v[j] > v[j + 1])

{

aux = v[j];

v[j] = v[j + 1];

v[j + 1] = aux;

}

printf("Vectorul ordonat crescator : \n");

for (i = 0; i < dim; i++)

printf("%d ", v[i]);

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 6:

{

if (ok == 1)

{

printf("Introduceti un numar, cu care sa se verifice divizibilitatea elementelor din vector : ");

scanf("%d", &nr);

printf("\nElementele divizibile cu %d : ", nr);

for (i = 0; i < dim; i++)

if (v[i] % nr == 0)

printf("%d ", v[i]);

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 7:

{

if (ok == 1)

{

max = v[0];

for (i = 0; i < dim; i++)

if (v[i] > max)

max = v[i];

printf("Cel mai mare element din vector : %d\n", max);

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 8:

{

if (ok == 1)

{

min = v[0];

for (i = 0; i < dim; i++)

if (v[i] < min)

min = v[i];

printf("Cel mai mic element din vector : %d\n", min);

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 9:

{

if (ok == 1)

{

max = v[0];

for (i = 0; i < dim; i++)

if (v[i] > max)

max = v[i];

for (i = 0; i < dim; i++)

if (v[i] > 0)

{

min = v[i];

break;

}

for (i = 0; i < dim; i++)

if ((v[i] < min) && (v[i] > 0))

min = v[i];

printf("Cel mai mic element pozitiv \* cel mai mare element = %d\n", (min\*max));

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 10:

{

if (ok == 1)

{

s = 0;

for (i = 0; i < dim; i++)

s += v[i];

printf("Media aritmetica a elementelor vectorului = %.2f\n", ((float)s / (float)(dim/sizeof(int))));

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 11:

{

if (ok == 1)

{

printf("Introduceti un numar, care sa fie adaugat la toate elementele pare : ");

scanf("%d", &nr);

for (i = 0; i < dim; i++)

if (w[i] % 2 == 0)

w[i] += nr;

printf("Vectorul nou-format : ");

for (i = 0; i < dim; i++)

printf("%d ", w[i]);

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

default:

printf("Eroare ! Optiune nedefinita !\n");

break;

}

}

printf("\n");

return 0;

}#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

void \_menu(void)

{

printf("\n------------------Meniu Interactiv------------------\n"

"0. Iesire\n"

"2. Afisarea elementelor vectorului\n"

"3. Afisarea sumei elementelor pare ale vectorului\n"

"4. Afisarea produsului elementelor impare ale vectorului\n"

"5. Afisarea elementelor vectorului in ordine crescatoare\n"

"6. Afisarea elementelor vectorului care sunt divizibile cu un numar\n"

"7. Afisarea elementului cel mai mare din vector\n"

"8. Afisarea elementului cel mai mic\n"

"9. Calculul produsului dintre elementul pozitiv cel mai mic cu cel mai mare\n"

"10. Afisarea mediei aritmetice\n"

"11. Adunarea unui numar la toate elementele pare a vectorului, apoi afisarea noului vector\n\n"

"Optiunea dvs : ");

}

int main(void)

{

int opt, ok = 1, dim, i, j, s, p, aux, nr, min, max;

printf("Introduceti dimensiunea/ordinul vectorului : ");

scanf("%d", &dim);

int\* v = (int\*)malloc(dim \* sizeof(int));

for (i = 0; i < dim; i++)

{

printf("vector[%d] : ", i);

scanf("%d", v + i);

}

printf("\n");

while (1)

{

\_menu();

scanf("%d", &opt);

printf("\n");

if (opt == 0)

exit(0);

switch (opt)

{

case 2:

{

if (ok == 1)

{

printf("Elementele vectorului : ");

for (i = 0; i < dim; i++)

{

printf("%d ", \*(v+i));

}

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 3:

{

if (ok == 1)

{

s = 0;

for (i = 0; i < dim; i++)

if ((\*(v+i) & 1) == 0)

s += \*(v+i);

printf("Suma elementelor pare : %d\n", s);

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 4:

{

if (ok == 1)

{

p = 1;

for (i = 0; i < dim; i++)

if ((\*(v+i) & 1) == 1)

p \*= \*(v+i);

printf("Produsul elementelor impare : %d\n", p);

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 5:

{

if (ok == 1)

{

for (i = 0; i < dim - 1; i++)

for (j = 0; j < dim - i - 1; j++)

if (\*(v+j) > \*(v+j+1))

{

aux = \*(v+j);

\*(v+j) = \*(v+j+1);

\*(v+j+1) = aux;

}

printf("Vectorul ordonat crescator : \n");

for (i = 0; i < dim; i++)

printf("%d ", \*(v+i));

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 6:

{

if (ok == 1)

{

printf("Introduceti un numar, cu care sa se verifice divizibilitatea elementelor din vector : ");

scanf("%d", &nr);

printf("\nElementele divizibile cu %d : ", nr);

for (i = 0; i < dim; i++)

if (\*(v+i) % nr == 0)

printf("%d ", \*(v+i));

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 7:

{

if (ok == 1)

{

max = \*(v+0);

for (i = 0; i < dim; i++)

if (\*(v+i) > max)

max = \*(v+i);

printf("Cel mai mare element din vector : %d\n", max);

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 8:

{

if (ok == 1)

{

min = \*(v+0);

for (i = 0; i < dim; i++)

if (\*(v+i) < min)

min = \*(v+i);

printf("Cel mai mic element din vector : %d\n", min);

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 9:

{

if (ok == 1)

{

max = \*(v+0);

for (i = 0; i < dim; i++)

if (\*(v+i) > max)

max = \*(v+i);

for (i = 0; i < dim; i++)

if (\*(v+i) > 0)

{

min = \*(v+i);

break;

}

for (i = 0; i < dim; i++)

if ((\*(v+i) < min) && (\*(v+i) > 0))

min = \*(v+i);

printf("Cel mai mic element pozitiv \* cel mai mare element = %d\n", (min \* max));

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 10:

{

if (ok == 1)

{

s = 0;

for (i = 0; i < dim; i++)

s += \*(v+i);

printf("Media aritmetica a elementelor vectorului = %.2f\n", ((float)s / (float)(dim / sizeof(int))));

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

case 11:

{

if (ok == 1)

{

printf("Introduceti un numar, care sa fie adaugat la toate elementele pare : ");

scanf("%d", &nr);

for (i = 0; i < dim; i++)

if (\*(v+i) % 2 == 0)

\*(v+i) += nr;

printf("Vectorul nou-format : ");

for (i = 0; i < dim; i++)

printf("%d ", \*(v+i));

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea vectorului mai intai !\n");

}

break;

default:

printf("Eroare ! Optiune nedefinita !\n");

break;

}

}

free(v);

printf("\n");

return 0;

}

1. Să se scrie un program în C care, folosind un meniu interactiv să realizeze următoarele opțiuni: 0. Ieșire 1. Citirea unui vector de n caractere 2. Afișarea elementelor vectorului 3. Afișarea elementelor vectorului în ordine alfabetică 4. Afișarea elementelor vectorului în ordine invers alfabetică 5. Transformarea literelor mari în litere mici 6. Afișarea vocalelor conținute în vector

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

void \_menu(void)

{

printf("\n------------------Meniu Interactiv------------------\n"

"0. Iesire\n"

"1. Citirea unui vector de n caractere\n"

"2. Afisarea elementelor vectorului\n"

"3. Afisarea elementelor vectorului in ordine alfabetica\n"

"4. Afisarea elementelor vectorului in ordine invers alfabetica\n"

"5. Transformarea literelor mari in litere mici\n"

"6. Afisarea vocalelor continute in vector\n\n"

"Optiunea dvs : ");

}

int main(void)

{

int opt, ok, dim, i, j;

char v[100], aux, c1, c2;

while (1)

{

\_menu();

scanf("%d", &opt);

printf("\n");

if (opt == 0)

exit(0);

switch (opt)

{

case 1:

{

ok = 1;

printf("Introduceti dimensiunea/ordinul vectorului : ");

scanf("%d", &dim);

for (i = 0; i < dim; i++)

{

printf("vector[%d] : ", i);

scanf(" %c", &v[i]);

}

printf("\n");

}

break;

case 2:

{

if (ok == 1)

{

for (i = 0; i < dim; i++)

{

printf("%c ", v[i]);

}

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 3:

{

if (ok == 1)

{

for (i=0; i<dim-1; i++)

for (j = 0; j < dim - i - 1; j++)

{

c1 = v[j];

c2 = v[j + 1];

if (c1 <= 90)

c1 += 32;

if (c2 <= 90)

c2 += 32;

if (c1 > c2)

{

aux = v[j];

v[j] = v[j + 1];

v[j + 1] = aux;

}

}

printf("Vectorul ordonat alfabetic : \n");

for (i = 0; i < dim; i++)

printf("%c ", v[i]);

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 4:

{

if (ok == 1)

{

for (i = 0; i < dim - 1; i++)

for (j = 0; j < dim - i - 1; j++)

{

c1 = v[j];

c2 = v[j + 1];

if (c1 >= 97)

c1 -= 32;

if (c2 >= 90)

c2 -= 32;

if (c1 < c2)

{

aux = v[j];

v[j] = v[j + 1];

v[j + 1] = aux;

}

}

printf("Vectorul ordonat invers alfabetic : \n");

for (i = 0; i < dim; i++)

printf("%c ", v[i]);

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 5:

{

if (ok == 1)

{

for (i = 0; i < dim; i++)

if (v[i] <= 90)

v[i] += 32;

printf("Vectorul rezultant cu majusculele transformate in minuscule : \n");

for (i = 0; i < dim; i++)

printf("%c ", v[i]);

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 6:

{

if (ok == 1)

{

printf("Vocalele continute in vector : ");

for (i = 0; i < dim; i++)

if ((v[i] == 65) || (v[i] == 65 + 32) || (v[i] == 69) || (v[i] == 69 + 32) || (v[i] == 73) || (v[i] == 73 + 32) || (v[i] == 79) || (v[i] == 79 + 32) || (v[i] == 85) || (v[i] == 85 + 32))

printf("%c ", v[i]);

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

default:

printf("Eroare ! Optiune nedefinita !\n");

break;

}

}

printf("\n");

return 0;

}

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

void \_menu(void)

{

printf("\n------------------Meniu Interactiv------------------\n"

"0. Iesire\n"

"2. Afisarea elementelor vectorului\n"

"3. Afisarea elementelor vectorului in ordine alfabetica\n"

"4. Afisarea elementelor vectorului in ordine invers alfabetica\n"

"5. Transformarea literelor mari in litere mici\n"

"6. Afisarea vocalelor continute in vector\n\n"

"Optiunea dvs : ");

}

int main(void)

{

int opt, ok=1, dim, i, j;

char aux, c1, c2;

printf("Introduceti dimensiunea/ordinul vectorului : ");

scanf("%d", &dim);

char\* v = (int\*)malloc(dim \* sizeof(char));

for (i = 0; i < dim; i++)

{

printf("vector[%d] : ", i);

scanf(" %c", (v + i));

}

printf("\n");

while (1)

{

\_menu();

scanf("%d", &opt);

printf("\n");

if (opt == 0)

exit(0);

switch (opt)

{

case 2:

{

if (ok == 1)

{

for (i = 0; i < dim; i++)

{

printf("%c ", \*(v+i));

}

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 3:

{

if (ok == 1)

{

for (i = 0; i < dim - 1; i++)

for (j = 0; j < dim - i - 1; j++)

{

c1 = \*(v+j);

c2 = \*(v+j+1);

if (c1 <= 90)

c1 += 32;

if (c2 <= 90)

c2 += 32;

if (c1 > c2)

{

aux = \*(v+j);

\*(v+j) = \*(v+j+1);

\*(v+j+1) = aux;

}

}

printf("Vectorul ordonat alfabetic : \n");

for (i = 0; i < dim; i++)

printf("%c ", \*(v+i));

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 4:

{

if (ok == 1)

{

for (i = 0; i < dim - 1; i++)

for (j = 0; j < dim - i - 1; j++)

{

c1 = \*(v+j);

c2 = \*(v+j+1);

if (c1 >= 97)

c1 -= 32;

if (c2 >= 90)

c2 -= 32;

if (c1 < c2)

{

aux = \*(v+j);

\*(v+j) = \*(v+j+1);

\*(v+j+1) = aux;

}

}

printf("Vectorul ordonat invers alfabetic : \n");

for (i = 0; i < dim; i++)

printf("%c ", \*(v+i));

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 5:

{

if (ok == 1)

{

for (i = 0; i < dim; i++)

if (\*(v+i) <= 90)

\*(v+i) += 32;

printf("Vectorul rezultant cu majusculele transformate in minuscule : \n");

for (i = 0; i < dim; i++)

printf("%c ", \*(v+i));

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

case 6:

{

if (ok == 1)

{

printf("Vocalele continute in vector : ");

for (i = 0; i < dim; i++)

if ((v[i] == 65) || (v[i] == 65 + 32) || (v[i] == 69) || (v[i] == 69 + 32) || (v[i] == 73) || (v[i] == 73 + 32) || (v[i] == 79) || (v[i] == 79 + 32) || (v[i] == 85) || (v[i] == 85 + 32))

printf("%c ", \*(v+i));

printf("\n");

}

else

printf("Apelati optiunea 1 pentru introducerea matricei mai intai !\n");

}

break;

default:

printf("Eroare ! Optiune nedefinita !\n");

break;

}

}

free(v);

printf("\n");

return 0;

}