```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h> /* コンパイル時に-lmオプションが必要 */

#define N 6 /* 要素の数 */

double *dvector(int i, int j);  /* ベクトル領域の確保 */
void free_dvector(double *a, int i); /* 領域の解放 */

/* 1 ノルムの計算 a[m...n] */
double vector_norm1(double *a, int m, int n);
/* 2 ノルムの計算 a[m...n] */
double vector_norm2(double *a, int m, int n);
/* 比較関数 */
int double_comp(const void *s1, const void *s2);
/* 最大値ノルムの計算 a[m...n] */
double vector_norm_max(double *a, int m, int n);

int main(void)
{
        int i;  double *a;

        /* ベクトルの定義, 配列 a の添字は 1〜N */
        a  = dvector(1,N);
        for(i = 1; i <= N; i++) a[i]  = (double)(10 - i) / 20.0 * pow(-1.0,i);
        for(i = 1; i <= N; i++) printf("a[%d]=%f\n", i, a[i]);

        printf("ベクトル a の 1 ノルムは%f です\n", vector_norm1(a, 1, N));
        printf("ベクトル a の 2 ノルムは%f です\n", vector_norm2(a, 1, N));
        printf("ベクトル a の最大値ノルムは%f です\n", vector_norm_max(a, 1, N));
        free_dvector(a,1);  /* 領域の解放 */
        return 0;
}

double *dvector(int i, int j) /* a[i]〜a[j] の領域を確保 */
{
        double *a;

        if ((a = malloc(((j - i + 1) * sizeof(double)))) == NULL)
        {
                printf("メモリが確保できません (from dvector) \n");
                exit(1);
        }

        return (a - i);
}

void free_dvector(double *a, int i)
{
        free((void *)(a + i)); /* (void *) 型へのキャストが必要 */
}

/* 1 ノルムの計算 a[m...n] */
double vector_norm1(double *a, int m, int n)
{
        int i;
        double norm = 0.0;
        for(i = m; i <= n; i++){
                norm += fabs(a[i]);
        }
        return norm;
}

/* 2 ノルムの計算 a[m...n] */
double vector_norm2(double *a, int m, int n)
{
        int i;
        double norm  = 0.0;
        for(i = m; i <= n; i++){
                norm += a[i] * a[i];
        }
        norm = sqrt(norm);
        return norm;
}

/* 比較関数 (昇順) */
int double_comp(const void *s1, const void *s2)
{
        const double a1 = *((double *)s1); /* (double *) へキャスト */
        const double a2 = *((double *)s2); /* (double *) へキャスト */
        if (a1 < a2)
        {
                return -1;
        }
        else if (a1 == a2)
        {
                return 0;
        }
        else
        {
                return 1;
        }
}

/* 最大値ノルムの計算 a[m...n] */
double vector_norm_max(double *a, int m, int n)
{
        int i, tmp;
        tmp = n - m + 1; /* 全要素数の計算 */
        for(i = m; i <= n; i++) a[i] = fabs(a[i]);
        /* 並べ換え:先頭アドレスが m だけずれていることに注意 */
        qsort(a + m, tmp, sizeof(a[0]), double_comp);
        return a[n];
}
```