```
int main(int argc , char *argv[]){
   printf ("Hello world!");
   return 0;
}

float Q_rsqrt(float number ){
   long i ;
   float x2, y;
   const float threehalfs = 1.500000;
   x2 = number * 0.500000;
   y = number;
   i = *(long *)&y;
   i = 0 - (i >> 1);
   y = *(float *)&i;
   y = y * (threehalfs - (x2 * y * y));
   return y;
}

void quick_sort(int arr [20], int low , int high ){
   int pivot, j, temp, i;
   if (low < high){
      pivot = low;
      i = low;
      j = high;
      while (i < j){
         while ((arr[i] ≤ arr[pivot]) ∧ (i < high)){
            i++;
         }
         while (arr[j] > arr[pivot]){
            j–;
         }
         if (i < j){
            temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
         }
      }
      temp = arr[pivot];
      arr[pivot] = arr[j];
      arr[j] = temp;
      quick_sort(arr, low, j - 1);
      quick_sort(arr, j + 1, high);
   }
}

void duff(register short *to, register short *from, register count ){
```

```
    register n = (count + 7) / 8;
    switch (count % 8){
      case 0: do {
        to++ = *from++;
        case 7: *to++ = *from++;
        case 6: *to++ = *from++;
        case 5: *to++ = *from++;
        case 4: *to++ = *from++;
        case 3: *to++ = *from++;
        case 2: *to++ = *from++;
        case 1: *to++ = *from++;
      }
      while (–n > 0);
    }
}

send(to, from, count)register short *to, *from;
register count ;
{
    register n = (count + 7) / 8;
    switch (count % 8){
      case 0: do {
        to = *from++;
        case 7: *to = *from++;
        case 6: *to = *from++;
        case 5: *to = *from++;
        case 4: *to = *from++;
        case 3: *to = *from++;
        case 2: *to = *from++;
        case 1: *to = *from++;
      }
      while (–n > 0);
    }
}
```