Problem 1

(a)

$$\begin{aligned} P(U \le r(x)) &= E(I(U \le r(x))) = E(E(I(U \le r(x))|x)) = E(r(x)) \\ &= \int \frac{q(x)}{\alpha g(x)} g(x)dx = \frac{1}{\alpha}\int q(x)dx \end{aligned}$$

(b)

$$\begin{aligned} P(x \in A, U \le r(x)) &= E(I(U \le r(x))I(x \in A)) = E(r(x)I(x \in A)) \\ &= \int_A \frac{q(x)}{\alpha g(x)} g(x)dx = \frac{1}{\alpha}\int_A q(x)dx \end{aligned}$$

$$P(Y \in A) = P(x \in A|U \le r(x)) = \frac{P(x \in A, U \le r(x))}{P(U \le r(x))} = \int_A \frac{q(x)}{\int q(x)dx}dx$$

Problem 2

(a)

$$\int (2x^{\theta-1}e^{-x} + x^{\theta-1/2}e^{-x})dx = 2\Gamma(\theta) + \Gamma(\theta + \frac{1}{2}) \tag{1}$$

(b)

$$\begin{aligned} g(x) &= \omega_1 g_1(x) + \omega_2 g_2(x) \\ &= \frac{2\Gamma(\theta)}{2\Gamma(\theta) + \Gamma(\theta + \frac{1}{2})} \left(\frac{1}{\Gamma(\theta)} x^{\theta-1}e^{-x}\right) + \frac{\Gamma(\theta + \frac{1}{2})}{2\Gamma(\theta) + \Gamma(\theta + \frac{1}{2})} \left(\frac{1}{\Gamma(\theta + 1/2)} x^{\theta-1/2}e^{-x}\right) \end{aligned}$$

(c) Algorithm:

1. Sample $U \sim Ber\left(\frac{2\Gamma(\theta)}{2\Gamma(\theta)+\Gamma(\theta+\frac{1}{2})}\right)$;
2. If $U = 1$, sample $X \sim Gamma(1, \theta)$; if $U = 0$, sample $X \sim Gamma(1, \theta + 1/2)$. Then, X is a sample from $g(x)$.

(d) It can be shown that $f(x) \le g(x)$ for $x > 0$.
Algorithm:

1. Sample $X \sim g(x)$ and $U \sim Unif(0, 1)$ independently;
2. If $U > \frac{f(x)}{g(x)}$, then go to Step 1; otherwise, return X. The returned value is a random variable from $f(x)$.

Problem 3

C code:

```
#include<stdio.h>

#define N 16 /* number of observations */
#define P 2 /* number of predictors */

void dgesv_(int *n, int *NRHS, double *A, int *LDA,
            int *IPIV, double *B, int *LDB, int *INFO);

void dgemm_(char *TRANSA, char *TRANSB, int *M, int *n, int *K,
            double *ALPHA, double *A,nt *LDA, double *B,
            int *LDB, double *BETA, double *C, int *LDC);

int main(){
        double Y[N] = {60.323,61.122,60.171,61.187,63.221,
                63.639,64.989,63.761,66.019,67.857,68.169,
                66.513,68.655,69.564,69.331,70.551};
        double X[N][P] ={{83,107.608},{88.5,108.632},
                {88.2,109.773},{89.5,110.929},
                {96.2,112.075}, {98.1,113.27},
                {99,115.094},{100,116.219},
                {101.2,117.388},{104.6,118.734},
                {108.4,120.445},{110.8,121.95},
                {112.6,123.366},{114.2,125.368},
                {115.7,127.852},{116.9,130.081}};

        double x[N*(P+1)],xt[(P+1)*N],Design_Mat[(P+1)*(P+1)];
        int i,j,k;
        int n1=N,n2=P+1,n3=1,info, ipiv[P+1];
        double coef[P+1], beta=0, alpha=1;
        char TRANSA = 'N', TRANSB='N';

        // Add intercept column to X and X's transpose
        for(i=0;i<N;i++){
                for(j=0;j<=P;j++){
                        if(j==0){
                                x[i*(P+1)+j] = 1;
                               xt[j*N+i] = 1;
                        }
```

```
                        else{
                                x[i*(P+1)+j] = X[i][j-1];
                                xt[j*N+i] = X[i][j-1];
                        }
                }
        }

        dgemm_(&TRANSA, &TRANSB, &n2, &n2, &n1, &alpha, x,
                        &n2, xt, &n1, &beta,Design_Mat,&n2);

        dgemm_(&TRANSA, &TRANSB, &n2, &n3, &n1, &alpha, x,
                        &n2, Y, &n1, &beta,coef,&n2);

        dgesv_(&n2,&n3, Design_Mat,&n2,ipiv,coef,&n2,&info);

        printf("The regression coefficients: ");
        for(i=0;i<=P;i++)
                printf("%f  ", coef[i]);
        printf("\n");
        return(0);
}
```

Problem 4

C code:

```
#include<stdio.h>

#define N 16 /* number of observations */
#define P 2 /* number of predictors */

void dgesvd_(char *JOBU, char *JOBVT, int *M, int *n,
                double *A, int *LDA, double *S, double *U,
                int *LDU, double *VT, int *LDVT, double *WORK,
                int *LWORK, int *INFO);

int main(){
        double X[N][P] ={{83,107.608},{88.5,108.632},
                {88.2,109.773},{89.5,110.929},
                {96.2,112.075}, {98.1,113.27},
                {99,115.094},{100,116.219},
```

```
                {101.2,117.388},{104.6,118.734},
                {108.4,120.445},{110.8,121.95},
                {112.6,123.366},{114.2,125.368},
                {115.7,127.852},{116.9,130.081}};

        int i,j,k;
        int n1=N,n2=P,info,LWORK=5*N*P;
        char JOBU='A', JOBVT='A';
        double Zt[N*P],X_colMean[P],S[P],U[N*N],VT[N*N],WORK[5*N*P];

        for(i=0;i<P;i++){
                X_colMean[i] = 0;
                for(j=0;j<N;j++)
                        X_colMean[i] += X[j][i];
                X_colMean[i] = X_colMean[i]/N;
        }

        for(i=0;i<N;i++){
                for(j=0;j<P;j++){
                        X[i][j] = X[i][j] - X_colMean[j];
                        Zt[j*N+i] = X[i][j];
                }
        }

        dgesvd_(&JOBU,&JOBVT,&n1,&n2,Zt,&n1,S,U,&n1,
                        VT,&n2,WORK,&LWORK,&info);

        printf("The principal component scores:\n");
        for(i=0;i<N;i++){
                for(j=0;j<P;j++){
                        printf("%f ",U[j*N+i]*S[j]);
                }
                printf("\n");
        }

        return(0);
}
```

Problem 5

C code:

```
#include <stdio.h>
#include <math.h>
#define N 10 /*The length of array*/

int main(){
        double x[N] = {3.1, -1.2, 5.3, 1, 4.4, 21, 3, 7, -1.2, 3.2};
        double tmp;
        int i,j;

        for(i=0;i<N-1;i++){
                tmp = x[i+1];
                for(j=i;j>=0;j--){
                        if(x[j]>tmp){
                                x[j+1] = x[j];
                                if(j==0)
                                        x[j] = tmp;
                        }
                        else{
                                x[j+1] = tmp;
                                break;
                        }
                }
        }

        printf("Sorted data:\n");
        for(i=0;i<N;i++)
                printf("%f ", x[i]);
        printf("\nMedian:\n");

        if(fmod(N,2)<0.5){
                printf("%f", (x[N/2-1]+x[N/2])/2);
        }
        else{
                printf("%f",x[(N-1)/2]);
        }

        printf("\n");
```

Problem 5 continued on next page...

```
        return(0);
}
```