Student Take Home Examination (LTCA) (COMP6112036 - Algorithm and Programming) (Mubarak - 2802667545)

1. question_1.c

```c
#include <stdio.h>

int
main ()
{
int a, b, c;

printf ("Masukkan bilangan bulat pertama: ");
scanf ("%d", &a);
printf ("Masukkan bilangan bulat kedua: ");
scanf ("%d", &b);
printf ("Masukkan bilangan bulat ketiga: ");
scanf ("%d", &c);

if (a >= b && a >= c)
    {
    printf ("Bilangan terbesar adalah %d\n", a);
    }
else if (b >= a && b >= c)
    {
    printf ("Bilangan terbesar adalah %d\n", b);
    }
else
    {
    printf ("Bilangan terbesar adalah %d\n", c);
    }

return 0;
}
```

2. question_2.c

```c
#include <stdio.h>
int
main ()
{
int maxTime;
int i = 4;

printf ("Masukkan waktu maksimal: ");
scanf ("%d", &maxTime);
```

```
while (i <= maxTime)
    {
    printf ("%d ", i);
    i += 4;
    }

printf ("\n");
return 0;
}
```

3. question_3.c

```
#include <stdio.h>

int
main ()
{
float nilai, max;
float *ptr = &nilai;

printf ("Masukkan 5 nilai rasa: ");
for (int i = 0; i < 5; i++)
    {
    scanf ("%f", ptr);

    if (i == 0 || *ptr > max)
        {
        max = *ptr;
        }
    }

printf ("Skor rasa tertinggi: %.1f\n", max);
return 0;
}
```

4. question_4.c

```
#include <stdio.h>

int
factorial (int n)
{
if (n == 0 || n == 1)
    {
    return 1;
    }
else
    {
```

```
    return n * factorial (n - 1);
    }
}

int
main ()
{
int n;

printf ("Masukkan angka: ");
scanf ("%d", &n);

if (n < 0)
    {
    printf ("Tolong masukan bilangan bulat positif.\n");
    }
else
    {
    printf ("%d! = %d\n", n, factorial (n));
    }

return 0;
}
```

5. question_5.c

```
#include <ctype.h>
#include <stdio.h>
#include <string.h>

int
main ()
{
char kalimat[256];

printf ("Masukkan sebuah kalimat: ");
fgets (kalimat, sizeof (kalimat), stdin);

for (int i = 0; i < strlen (kalimat); i++)
    {
    kalimat[i] = tolower (kalimat[i]);
    }

printf ("Hasil: %s", kalimat);

return 0;
}
```

6. question_6.c

```c
#include <ctype.h>
#include <stdio.h>

int
main ()
{
FILE *file;
char ch;
int count = 0;

file = fopen ("angka.txt", "r");
if (file == NULL)
    {
    perror ("Gagal membuka file");
    return 1;
    }

while ((ch = fgetc (file)) != EOF)
    {
    if (isdigit (ch))
        {
        count++;
        }
    }

fclose (file);

printf ("Jumlah digit: %d\n", count);

return 0;
}
```

7. question_7.c

```c
#include <stdio.h>

int
sum (int a, int b)
{
return a + b;
}

int
main ()
{
int (*ptr) (int, int) = sum;
printf ("%d\n", ptr (3, 4));
}
```

output

```
7
```

8. question_8.c

```c
#include <stdio.h>
#include <string.h>

typedef struct
{
char name[32];
char dish[64];
int tasteScore;
int presentationScore;
int time;
} Peserta;

int
total (const Peserta *p)
{
return p->tasteScore + p->presentationScore;
}

void
showAll (const Peserta *arr, int n)
{
puts ("\nDaftar Peserta:");
for (int i = 0; i < n; i++)
    {
    printf ("%2d. %-10s - %-15s | Rasa:%3d | Penyajian:%3d | Total:%3d
| "
            "Waktu:%3d menit\n",
            i + 1, arr[i].name, arr[i].dish, arr[i].tasteScore,
            arr[i].presentationScore, total (&arr[i]), arr[i].time);
    }
puts ("");
}

typedef int (*Cmp) (const Peserta *, const Peserta *);
void
bubbleSort (Peserta *arr, int n, Cmp cmp)
{
int swapped = 1;
for (int pass = 0; pass < n - 1 && swapped; pass++)
    {
    swapped = 0;
    for (int i = 0; i < n - 1 - pass; i++)
        {
        if (cmp (&arr[i], &arr[i + 1]) > 0)
            {
            Peserta tmp = arr[i];
```

```
            arr[i] = arr[i + 1];
            arr[i + 1] = tmp;
            swapped = 1;
            }
        }
    }
}

int
cmpTotalDesc (const Peserta *a, const Peserta *b)
{
int ta = total (a), tb = total (b);
if (ta != tb)
    return (tb - ta);
return 0;
}

int
cmpTimeAsc (const Peserta *a, const Peserta *b)
{
return (a->time - b->time);
}

int
cmpNamaAsc (const Peserta *a, const Peserta *b)
{
return strcmp (a->name, b->name);
}

int
cmpTop3 (const Peserta *a, const Peserta *b)
{
int ta = total (a), tb = total (b);
if (ta != tb)
    return (tb - ta);
return (a->time - b->time);
}

void
top3 (const Peserta *src, int n)
{
Peserta temp[64];
if (n > 64)
    n = 64;
for (int i = 0; i < n; i++)
    temp[i] = src[i];

bubbleSort (temp, n, cmpTop3);

int m = n < 3 ? n : 3;
puts ("\nTop 3 Peserta Terbaik:");
for (int i = 0; i < m; i++)
    {
    printf ("%d. %s - %s - Nilai: %d - Waktu: %d menit\n", i + 1,
```

```
            temp[i].name, temp[i].dish, total (&temp[i]),
temp[i].time);
    }
puts ("");
}

void
rataRata (const Peserta *arr, int n)
{
if (n == 0)
    {
    puts ("Tidak ada peserta.");
    return;
    }
double sumR = 0, sumP = 0;
for (int i = 0; i < n; i++)
    {
    sumR += arr[i].tasteScore;
    sumP += arr[i].presentationScore;
    }
printf ("\nRata-rata Nilai Rasa: %.2f\n", sumR / n);
printf ("Rata-rata Nilai Penyajian: %.2f\n\n", sumP / n);
}

int
main (void)
{
Peserta peserta[] = { { "Adit", "Rendang", 95, 95, 45 },
                      { "Bella", "Salmon Steak", 92, 96, 50 },
                      { "Chika", "Laksa", 90, 95, 48 },
                      { "Dion", "Sate Ayam", 85, 88, 40 },
                      { "Eka", "Sop Buntut", 88, 86, 55 },
                      { "Fani", "Gudeg", 91, 80, 42 },
                      { "Gilang", "Rawon", 87, 84, 47 },
                      { "Hana", "Pempek", 89, 82, 49 } };
int n = (int)(sizeof (peserta) / sizeof (peserta[0]));

Peserta work[64];
for (int i = 0; i < n; i++)
    work[i] = peserta[i];

while (1)
    {
    puts ("~~~ Sistem Peringkat Master Chef Nasional ~~~");
    puts ("1. Tampilkan seluruh data peserta");
    puts ("2. Urutkan berdasarkan total nilai (desc)");
    puts ("3. Urutkan berdasarkan time memasak (asc)");
    puts ("4. Urutkan berdasarkan name peserta (asc)");
    puts ("5. Tampilkan 3 peserta terbaik (total tertinggi, tiebreak
time "
          "tercepat)");
    puts ("6. Tampilkan rata-rata nilai tasteScore &
presentationScore");
    puts ("7. Keluar");
```

```
    printf ("Pilihan (1-7): ");

    int pilihan;
    if (scanf ("%d", &pilihan) != 1)
        {
        int c;
        while ((c = getchar ()) != '\n' && c != EOF)
            {
            }
        puts ("Input tidak valid. Coba lagi.\n");
        continue;
        }
    if (pilihan < 1 || pilihan > 7)
        {
        puts ("Pilihan harus 1-7. Coba lagi.\n");
        continue;
        }

    if (pilihan == 7)
        {
        puts ("Terima kasih. Keluar program.");
        break;
        }

    switch (pilihan)
        {
        case 1:
        showAll (work, n);
        break;
        case 2:
        bubbleSort (work, n, cmpTotalDesc);
        puts ("\nDiurutkan berdasarkan total nilai (desc):");
        showAll (work, n);
        break;
        case 3:
        bubbleSort (work, n, cmpTimeAsc);
        puts ("\nDiurutkan berdasarkan time memasak (asc):");
        showAll (work, n);
        break;
        case 4:
        bubbleSort (work, n, cmpNamaAsc);
        puts ("\nDiurutkan berdasarkan name peserta (asc):");
        showAll (work, n);
        break;
        case 5:
        top3 (work, n);
        break;
        case 6:
        rataRata (work, n);
        break;
        }
    }

return 0;
```

```
}
```