

# Il prova in itinere di Calcolatori Elettronici – Ing. Informatica del 27/01/15

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

Tempo 2 h 45 min. Punteggi: Q1. 16 punti, Q2. 7 punti, Q3. 7 punti  
Per superare la prova è necessario avere un punteggio  $P \geq 18$  con  $Q1 \geq 10$

**Q1.** Creare un file di testo utilizzando la convenzione **cognome.nome.s** (es. *rossi.paolo.s*). )  
Scrivere un programma in linguaggio Assembly MIPS che traduce il seguente programma C:

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

int verifica(int Ai, int i)
{ int t;
  if(Ai < 10)
    t=i+1;
  else t=i;
  return t;
}

main() {
  char S[5][40];
  int A[5],B[5],N[5],num,i,j;

  i=0;
  do
  { printf("Inserire un numero < 10\n");
    scanf("%d",&A[i]);
    i=verifica(A[i],i);
  } while (i<5);

  for(i=0; i<5;i++) {
    printf("Inserisce una stringa con al massimo %d caratteri\n", A[i]);
    gets(S[i]);
    N[i]=strlen(S[i]);
  }

  for(i=0;i<5;i++) {
    num=0;
    for(j=0;j<N[i];j++)
      if(S[i][j] < 58)
        num++;
    B[i]=num;
  }

  for(i=0;i<5;i++)
    printf("B[%d]= %d\n", i,B[i]);
}
```

Q2. Discutere della realizzazione microprogrammata dell'unità di controllo mostrandone anche uno schema.

Q3. Discutere del problema del block placement nella gerarchia di memoria

# Il prova in itinere di Calcolatori Elettronici – Ing. Informatica del 27/01/15

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

Tempo 2 h 45 min. Punteggi: Q1. 16 punti, Q2. 7 punti, Q3. 7 punti  
Per superare la prova è necessario avere un punteggio  $P \geq 18$  con  $Q1 \geq 10$

**Q1.** Creare un file di testo utilizzando la convenzione **cognome.nome.s** (es. *rossi.paolo.s*). Scrivere un programma in linguaggio Assembly MIPS che traduce il seguente programma C:

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

int leggi(char *stringa)
{ int d;

  do {
    printf("inserisci una stringa con almeno 12 caratteri");
    gets(stringa);
    d=strlen(stringa);
  } while (d <12);
  return d;
}

main() {
  char stringa[40];
  int A[4],B[4],num,i;

  num=leggi(stringa);

  printf("La stringa ha %d caratteri\n",num);
  for(i=0;i<4;i++)
    A[i]=(stringa[i]-48)*2+(stringa[i+4]-48)+(stringa[i+8]-48)/2;

  for(i=0;i<4;i++)
    { printf("Inserisci un numero positivo\n");
      scanf("%d",&B[i]);
      if(A[i]-B[i] > 10)
        B[i]=B[i]+10;
      else if((A[i]+B[i])/2<10)
        B[i]=B[i]+5;
    }

  for(i=0;i<4;i++)
    printf("A[%d]= %d B[%d]= %d\n", i,A[i],i,B[i]);
}
```

**Q2.** Discutere dell'Instruction Set Architecture di un processore evidenziando i parametri su cui agire e le soluzioni che più diffusamente vengono utilizzate.

**Q3.** Discutere dell'accesso diretto alla memoria con l'ausilio di uno schematico

## Il prova in itinere di Calcolatori Elettronici – Ing. Informatica del 27/01/15

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

Tempo 2 h 45 min. Punteggi: Q1. 15 punti, Q2. 7 punti, Q3. 7 punti  
Per superare la prova è necessario avere un punteggio  $P \geq 18$  con  $Q1 \geq 10$

Q1. Creare un file di testo utilizzando la convenzione **cognome.nome.s** (es. *rossi.paolo.s*).  
Scrivere un programma in linguaggio Assembly MIPS che traduce il seguente programma C:

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

int verifica(int Ni, int Ai, int i)
{ int t;
    if(Ni >= Ai)
        t=i+1;
    else t=i;
    return t;
}

main() {
    char S[5][40];
    int A[5],B[5],N[5],num,i,j;

    i=0;
    do
    { printf("Inserire un numero >= 3\n");
      scanf("%d",&A[i]);
      if(A[i]>=3)
          i++;
    } while (i<5);

    i=0;
    do
    {printf("Inserisce una stringa con almeno %d caratteri\n", A[i]);
      gets(S[i]);
      N[i]=strlen(S[i]);
      i=verifica(N[i],A[i],i);
    } while(i<5);

    for(i=0;i<5;i++) {
        num=0;
        for(j=0;j<N[i];j++)
            num= num + S[i][j]-48;

        B[i]=num/N[i];
    }

    for(i=0;i<5;i++)
        printf("B[%d]= %d\n", i,B[i]);
}
```

Q3. Discutere della realizzazione microprogrammata dell'unità di controllo mostrandone anche uno schema.

Q4. Confrontare la tecnica delle interruzioni con quella del polling evidenziando anche i vantaggi e i problemi di ciascuna soluzione.

# Il prova in itinere di Calcolatori Elettronici – Ing. Informatica del 27/01/15

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

Tempo 2 h 45 min. Punteggi: Q1. 16 punti, Q2. 7 punti, Q3. 7 punti  
Per superare la prova è necessario avere un punteggio  $P \geq 18$  con  $Q1 \geq 10$

Q1. Creare un file di testo utilizzando la convenzione **cognome.nome.s** (es. *rossi.paolo.s*).  
Scrivere un programma in linguaggio Assembly MIPS che traduce il seguente programma C:

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

int calcola(char Sij, int Ni)
{ int t;
  t=Sij-48;
  if(t >= Ni)
    return t-(Ni/2);
  else if(t>= Ni/2)
    return t-Ni;
  else return t;
}

main() {
  char S[6][32];
  int A[6],N[6],num,i,j;

  for(i=0;i<6;i++)
  {
    printf("Inserisce una stringa \n");
    gets(S[i]);
    N[i]=strlen(S[i]);
  }

  for(i=0;i<6;)
  { do {
    printf("Inserire un numero positivo < %d\n",N[i]);
    scanf("%d",&j);
  } while(j>=N[i]);
  printf("S=%d\n", S[i][j]);
  A[i]= calcola(S[i][j],N[i]);
  if(A[i] > 0)
    i++;
  }

  for(i=0;i<6;i++)
    printf("A[%d]= %d\n", i,A[i]);
}
```

Q2. Discutere della realizzazione cablata dell'unità di controllo mostrandone anche uno schema.

Q3. Discutere delle possibili strategie in scrittura e di cosa accade nel caso di miss in scrittura per una memoria cache

# Il prova in itinere di Calcolatori Elettronici – Ing. Informatica del 27/01/15

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

Tempo 2 h 45 min. Punteggi: Q1. 16 punti, Q2. 7 punti, Q3. 7 punti  
Per superare la prova è necessario avere un punteggio  $P \geq 18$  con  $Q1 \geq 10$

Q1. Creare un file di testo utilizzando la convenzione **cognome.nome.s** (es. *rossi.paolo.s*).  
Scrivere un programma in linguaggio Assembly MIPS che traduce il seguente programma C:

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

int calcola(char *ST, int Ni)
{ int num,j;

  num=0;
  for(j=0;j<Ni;j++)
    if(ST[j] < 58)
      num++;
    else if (ST[j] < 116)
      num+=2;
  return num;
}

main() {
  char S[4][40];
  int A[4],B[4],N[4],num,i,j;

  i=0;
  do
  { printf("Inserire un numero < 15\n");
    scanf("%d",&A[i]);
    if(A[i]<15)
      i++;
  } while (i<4);

  for(i=0; i<4;i++) {
    printf("Inserisce una stringa con al massimo %d caratteri\n", A[i]);
    gets(S[i]);
    N[i]=strlen(S[i]);
  }

  for(i=0;i<5;i++)
    B[i]=calcola(S[i],N[i]);

  for(i=0;i<5;i++)
    printf("B[%d]= %d\n", i,B[i]);
}
```

Q2. Discutere dell'Instruction Set Architecture mostrando in particolare le soluzioni che più diffusamente vengono utilizzate per i processori RISC.

Q3. Descrivere la tecnica delle interruzioni vettorizzate, mostrando anche un possibile schematico .

# Il prova in itinere di Calcolatori Elettronici – Ing. Informatica del 27/01/15

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

Tempo 2 h 45 min. Punteggi: Q1. 16 punti, Q2. 7 punti, Q3. 7 punti  
Per superare la prova è necessario avere un punteggio  $P \geq 18$  con  $Q1 \geq 10$

Q1. Creare un file di testo utilizzando la convenzione **cognome.nome.s** (es. *rossi.paolo.s*). Scrivere un programma in linguaggio Assembly MIPS che traduce il seguente programma C:

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

int aggiorna(int s, int p, char c)
{ int t,r;

    if(c < 48)
        t=0;
    else if(c<58)
        t=c-48;
    else if(c<65)
        t=0;
    else if (c<71)
        t=c-55;
    else t=0;

    r= s+t*p;
    return r;
}

main() {
    char stringa[40];
    int A[4],B[4],d,p,i,j,s;

    for(i=0;i<4;i++)
    { printf("inserisci un numero esadecimale (0,1,..., A, ..F)");
      gets(stringa);
      d=strlen(stringa);

      s=0;
      p=1;
      for(j=d-1;j>=0;j--)
      { s= aggiorna(s,p,stringa[j]);
        p=p*16;
      }
      A[i]=s;

      printf("Inserisci un numero positivo\n");
      scanf("%d",&B[i]);
      B[i]=B[i]+A[i];
    }

    for(i=0;i<4;i++)
        printf("A[%d]= %d B[%d]= %d\n", i,A[i],i,B[i]);
}
```

Q2. Discutere della realizzazione microprogrammata dell'unità di controllo mostrandone anche uno schema.

Q3. Discutere del problema della block identification nella gerarchia di memoria

## Il prova in itinere di Calcolatori Elettronici – Ing. Informatica del 27/01/15

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

Tempo 2 h 45 min. Punteggi: Q1. 16 punti, Q2. 7 punti, Q3. 7 punti  
Per superare la prova è necessario avere un punteggio  $P \geq 18$  con  $Q1 \geq 10$

Q1. Creare un file di testo utilizzando la convenzione **cognome.nome.s** (es. *rossi.paolo.s*).  
Scrivere un programma in linguaggio Assembly MIPS che traduce il seguente programma C:

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

int verifica(char *V, int Ni, int Ai)
{ int j;

  if(Ni < Ai)
    { for(j=Ni;j<Ai;j++)
      V[j]=48;
      return Ai;
    }
  else return Ni;
}

main() {
  char S[5][40];
  int A[5],B[5],N[5],num,i,j;

  for(i=0;i<5;i++)
  { printf("Inserire un numero >=4 \n");
    scanf("%d",&A[i]);
    if(A[i]<4)
      A[i]=4;

    printf("Inserisce una stringa con almeno %d caratteri\n", A[i]);
    scanf("%s",S[i]);
    N[i]=strlen(S[i]);
    N[i]=verifica(S[i],N[i],A[i]);

    B[i]=48;
    for(j=0;j<N[i];j++)
      if(S[i][j]>B[i])
        B[i]=S[i][j];

    B[i]=B[i]-48;
  }

  for(i=0;i<5;i++)
    printf("B[%d]= %d\n", i,B[i]);
}
```

Q2. Discutere della valutazione delle prestazioni di un processore, come calcolarle e da quali parametri dipendono e come variano tali parametri in funzione delle scelte architetturali.

Q3. Discutere dell'accesso diretto alla memoria con l'ausilio di uno schematico.

# Il prova in itinere di Calcolatori Elettronici – Ing. Informatica del 27/01/15

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

Tempo 2 h 45 min. Punteggi: Q1. 16 punti, Q2. 7 punti, Q3. 7 punti  
Per superare la prova è necessario avere un punteggio  $P \geq 18$  con  $Q1 \geq 10$

Q1. Creare un file di testo utilizzando la convenzione **cognome.nome.s** (es. *rossi.paolo.s*).  
Scrivere un programma in linguaggio Assembly MIPS che traduce il seguente programma C:

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

int verifica_calcola(char *st, int Ni)
{
    int t,j;

    t=0;
    for(j=0;j<Ni;j++)
        if(st[j]<48)
            return 0;
        else if(st[j]>=58)
            return 0;
        else t=t+st[j]-48;

    return t;
}

main() {
    char S[6][32];
    int A[6],N[6],B[6],num,i,j;

    for(i=0;i<6;)
    {
        printf("Inserisce una stringa di soli numeri\n");
        gets(S[i]);
        N[i]=strlen(S[i]);
        A[i]=verifica_calcola(S[i],N[i]);
        if(A[i]!=0)
            i++;
        else printf("stringa errata\n");
    }

    for(i=0;i<6;i++)
    {
        printf("Inserire un numero positivo \n");
        scanf("%d",&num);
        B[i]=A[i]+num;
    }

    for(i=0;i<6;i++)
    {
        printf("A[%d]= %d\n", i,A[i]);
        printf("B[%d]= %d\n", i,B[i]);
    }
}
```

Q2. Discutere delle possibili strategie in scrittura e di cosa accade nel caso di miss in scrittura per una memoria cache.

Q3. Confrontare la tecnica delle interruzioni con quella dell'accesso diretto in memoria evidenziando anche i vantaggi e i problemi di ciascuna soluzione.